

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Patent Application Ser. No.: 09/466,627 Group Art Unit: 2176

Filing Date: 12/17/1999 Examiner: M. NGUYEN

Attorney Docket Number YO999-429 Inventor Name(s): LO ET AL.

Title: METHOD AND APPARATUS FOR CONVERTING BETWEEN DATA SETS AND
XML DOCUMENTS

Mail Stop Appeal Brief
Commissioner for Patents
P.O. Box 1450
Alexandria VA 22313-1450

APPEAL BRIEF

Sir:

I. REAL PARTY IN INTEREST

The real party in interest is IBM Corporation.

II. RELATED APPEALS AND INTERFERENCES

Applicant is not aware of any related appeals or interferences. There is, however, a co-pending petition from the refusal of the Examiner to enter the latest declaration under rule 131. That petition and the decision on the petition are included in the appendices.

III. STATUS OF CLAIMS

Claims 4-6, 13-15, 47, 50, 62, and 65 have been cancelled¹.

Claims 1-3, 7-12, 16-46, 48, 49, 51-61, 63, 64, and 66-96 stand rejected over US Pat No. 459 (“Chang”).

This is an appeal from the final rejection of Claims 1-3, 7-12, 16-46, 48, 49, 51-61, 63, 64, and 66-80, 84-96

IV. STATUS OF AMENDMENTS

There was no amendment under section 116, though there were remarks that were not considered persuasive.

V. SUMMARY OF CLAIMED SUBJECT MATTER

The invention is software and computer related. It has to do with converting between databases and XML. XML is a computer language used for generating documents, such as web pages.

The original claims of this application are organized behind 3 independent claims, a method claim (claim 1), a medium claim (claim 46), and a device claim (claim 61). The limitations of these independent claims are fairly analogous, i.e. they relate to establishing a mapping into XML elements and attributes from lists and scalars. Per page 4 of the spec, “for the purpose of this application a scalar is a single value and a list is a list of values.” The

¹ If the amendment after notice of appeal has been entered, then claims 81-83 are also cancelled.

preambles of these claims differ according to these three categories. Those claims depending from claim 1 are mostly organized into triplets. Each triplet includes: a method claim, a result of the method as stored on a medium, and a device that includes the medium and can use the result. Thus, in the original claims, there were fifteen groups each containing five closely related claims.

The numbering got confused because the claims were improvidently narrowed, then returned to their original scope; but essentially the original structure remains, with a few added claims.

General Explanation of the concepts of the invention

Fig. 2 — together with the accompanying text on p. 10 of spec — shows on the left side a data set 200 with a schema 201. On the right side is an XML document 204 with a Document Type Definition (“DTD”) 205. These four things existed prior to the invention. The invention 205 relates to interfacing between them.

In case the Board does not happen to know, XML is a language similar to HTML, used for generating Internet type documents. DTD’s are used in conjunction with XML documents. Data sets, such as relational databases, have been used for a fairly long time to store information. An example of a database is, for instance, the patent office online records relating to copies of issued patents. Databases have schema used to control their formatting. An example of a schema is shown at Fig. 3a. An example of a DTD is shown in Fig. 3b and explained at p. 12 of the specification.

Databases typically include scalars and lists. For the purpose of this application a scalar is a single value and a list is a list of values, per the specification at page 4, lines 5-6.

Fig. 4, see pp 13-14, schematically shows the concept of mapping from a database to XML. A mapping might go in either direction, as shown by the two headed arrow 402 in Fig. 4; however, the claims on appeal here only recite mapping TO XML.

Fig. 5 shows a bit of the preferred embodiment, an annotated DTD, also called DTDSA. A comparison of Fig. 3b with Fig. 5 reveals that the DTDSA has been changed from the DTD by the additions of annotations, e.g. 505, 510, 515, 520, 535, 540, 545. These annotations are in constructs of a mapping language. The annotations are inserted in the DTD by the engine shown in Fig. 6b at 605. The annotations are then used by an engine 705, in Fig. 7b to generate XML documents from the data sets.

Claims 1, 46, 61

Claim 1 recites a computer method. The method includes executing an operation in at least one data processing device.

Claim 1 recites establishing a mapping from lists and scalars corresponding to at least one data source into XML elements and attributes. The preferred embodiment of this is especially shown with respect to Figs. 6a and b explained at pp. 25-30.

Claims 46 and 61 are analogous, except that 46 is a medium claim and 61 is an apparatus claim.

Claims 2, 8, 11, 17, 20, 23, 26, 29, 32, 35, 38, 41, 44, 77, 88, 91

These claims each recite a medium embodying a result of a prior claim. For instance, claim 2 recites a medium embodying a result of claim 1. The claims of this group cannot be said to stand and fall together, because each depends from a distinct claim; however, they all relate to a medium embodying a mapping from lists and scalars into XML elements and attributes and all have very similar if not identical rejections. This is shown at 615' of Fig. 6b, together with Fig. 1, that shows a processor 104 and medium 106. Fig. 1 is discussed at pages 7-10 of the spec.

Claims 3, 9, 12, 18, 21, 24, 27, 30, 33, 36, 39, 42, 45, 78, 89, 92

These claims each depend on one of the claims in the group discussed immediately prior to this group. These claims relate to a device including a processor that is configured to use the medium of the parent claims to produce an XML document based on the mapping. Again, the claims of this group cannot be said to stand and fall together, because each depends from a distinct claim; however they all have very similar if not identical rejections. Producing an XML document from the mapping is shown at 715, Fig.s 7a and 7b.

Claims 10, 49, and 64

These claims recite expressing the mapping in constructs of a mapping language. The preferred embodiment of such constructs is shown, for instance, at Fig. 5, e.g. 505, 510, 515, 520, 535, 540, 545, as explained above in the general summary.

Claims 90, 94, 96

These claims recite an annotated DTD. This is a narrower recitation of what is shown in Fig. 5, as discussed in the previous paragraph.

Claims 76, 79, 80

These claims recite that an XML document generated from the annotated DTD is guaranteed to conform to the DTD. If one of ordinary skill in the art follows the discussion in the disclosure, he or she will be able to verify that resulting XML is guaranteed to conform to the DTD, see especially page 10, Lines 14-15 and page 19, Line 20-21.

Claims 16, 51, 66

These claims recite that the constructs comprise at least one of a value specification and a binding specification, see pp. 14-17 of the specification. The constructs are recited as in a DTD in the claims from which this group depends. That limitation is discussed elsewhere, see e.g. claim 90.

Claims 19, 52, 67

These claims recite that the construct comprises a parameter. At least one of parameters is determinable at the time of generation of at least one XML element.

The specification is replete with instances of parameters, for instance at Fig. 5, at 505, in the expression `poid(r)`, where “r” is a parameter, further discussed at p. 41 at lines 8-10. The generation of XML documents, which includes determining parameters, is discussed starting at p.

30.

Claim 21

The added limitation of passing of the parameters here in this claim is shown in several places, for instance, figures 8, 9, and p. 39, line 19 through p. 41.

Claims 25, 54, 69

These claims recite associating one or more lists of data objects or formulas producing data objects with each DTD construct having a repetition symbol at the end, see e.g. p. 23, ll. 14-21. For each of the claims in this group, the definition of “DTD construct” is incorporated from a prior claim, where the constructs are recited as inserted into a DTD to create an annotated DTD.

Claim 27

For the added limitation of “associating” see p. 25, line 12, *et seq.*

Claims 28, 55, 70

These claims recite associating one or more lists of data objects or formulas producing data objects with each DTD construct which is not a #PCDATA, a choice list, or an attribute list, and does not end with a repetition symbol, see e.g. p. 21, ll 6-15, p. 26 ll. 3-7. Again, for each of the claims in this group, the definition of “DTD construct” is incorporated from a prior claim, where the constructs are recited as inserted into a DTD to create an annotated DTD.

It can be seen from the text of the specification that the claims of this group use a slightly abbreviated wording to describe what is happening. Actually, the DTD construct is *related* to the #PCDATA, choice list, or attribute list. The PCDATA#, choice list, or attribute list is from the DTD itself, rather than from the construct. If the Board prefers, Applicants could clarify this point with an amendment; however, Applicants believe that one of ordinary skill in the art would understand what is meant here based on the language of the claims and the underlying discussion in the specification.

Claims 34, 57, 72

These claims recite

- first associating one or more lists of data objects or formulas producing data objects with a DTD construct;
- second associating at least one of the lists or formulas with at least one variable name; and
- using the variable name as a parameter in at least one other formula.

Please see e.g. drawing items 515 and 535. This claim cannot be located to a single part of the disclosure, and can perhaps be viewed as covering the broad concept of the disclosure starting at p. 14, particularly the “Establishing a Mapping” section starting at p. 25.

It should be noted again that, given the dependent nature of these claims, the reader must look to earlier claims for the definition of the term “DTD construct”, i.e. as relating to an annotation to a DTD.

Claim 37, 58, 73

These claims recite associating at least one environment with an XML element. It is to be noted that “environment” is defined in the specification at page 31, i.e. ‘a set of variable/value pairs called the “environment.”’

Claims 40, 59, 74

These claims recite that

- information from a parent XML element of the respective XML element; and
- information from a binding specification of a DTD construct associated with the respective XML element.

More information about this concept can be found at pp. 18-19 about ‘ancestral’ or parent relations between DTD constructs; and pp. 25-26, especially p. 26, line 20 where “higher level binding spec” means ancestor binding spec.

Claims 43, 60, 75

These claims recite

- the mapping includes at least one respective specification corresponding to at least one respective XML element;
- the specification comprises at least one parameter for receiving a value upon generation of an XML document; and

- the method further comprises, upon generation of an XML document, sending the at least one parameter a value according to at least one variable/value pair in the at least one respective environment

More information about these recitations can be found in the specification at p. 30, line 6 through page 31, line 14. Again, these are dependent claims, so the definition of the term “mapping” must be taken from claim one.

Claims 84, 85, and 86

These claims recite that the mapping is responsive to a user mapping specification. This is supported in the specification at p. 28, line 10 and Fig. 6A, element 612’

Claims 87 , 93. 95

These claims recite that there are at least two data sources and the sources are of different types, see e.g. p. 10, lines 3-5.

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

The rejections over Chang are to be reviewed.

VII. THE ARGUMENT

General comments about terminology in the Chang reference

Applicants note one slightly confusing aspect of Chang. Chang repeatedly uses the acronym “DB2XML.” Many people might mistakenly read this acronym as “data base to XML;” however, if one looks carefully at the Chang patent, one can see that this is not what Chang means. At col. 8, line 11, Chang explains that DB2[®] is the name of a particular type of database, and that the patent relates to XML documents stored in a DB2 database. Therefore, for Chang, “DB2XML” means a DB2 type database containing XML documents. Applicants respectfully submit that reading “DB2XML” from Chang to mean “data base to XML” or to imply conversion from database to XML constitutes impermissible hindsight in light of Applicants’ disclosure.

Argument in advisory action

The advisory action asks Applicant to review a section of the background of the invention in the reference at col. 2, lines 61-4. This section of the reference does not refer to a computer method that includes an operation to be executed, where that operation is establishing a mapping between lists and scalars and XML elements and attributes. Instead, this section of the reference relates to conventional manual programming of XML documents. Accordingly, the section fails to overcome the previously presented arguments for validity of the claims.

Claim 1, 46, 61

Claim 1 recites establishing a mapping from lists and scalars corresponding to at least one data source into XML elements and attributes. In these claims, the lists and scalars are mapped

to XML elements and attributes. The XML elements are not themselves the lists and scalars in the data source, as defined in the spec. To the extent that the reader may not understand this recitation, the reader needs to refer to the meaning of the terminology “mapping ... into” in the specification. The specification makes clear that this terminology relates to retrieving data from a relational database into an XML document. The lists and scalars are part of a source that needs to be made compatible with XML.

In reading over the reference, the undersigned is just not finding this. The reference talks about a database of XML documents, an extender which describes the XML documents, and DTD's for the XML documents. There is metadata for the XML documents in the database. Data appears to be retrieved out of the database of XML documents. Assuming *arguendo* that there is some mapping it would be *from* XML documents, but not *to* XML elements and attributes. Applicants accordingly respectfully submit that the Examiner mischaracterizes the reference.

Claims 2, 8, 11, 17, 20, 23, 26, 29, 32, 35, 38, 41, 44, 77, 88, 91

Some of these claims will be discussed separately below. However, with respect to all these claims, the Examiner has cited Chang's computer-readable medium, col. 6, line 59 and one or more processors, col. 6, line 21-22.

Applicants do not claim the general concept of a medium or a processor. Applicants claim a medium embodying a result of a method, namely a product by process claim. The method claims recite establishing a mapping from lists and scalars to XML elements and attributes. Accordingly, the medium is embodying the mapping.

The portion of the reference cited by the Examiner for the medium recites software called RDBMS and described in lines 46-53. This software appears to be database management software, per col. 2, lines 30-31 of the reference. Applicants find no teaching or suggestion in the reference that a mapping between lists and scalars and XML elements and attributes is embodied on a computer readable medium. Applicants accordingly respectfully submit that the Examiner has mischaracterized the reference.

Claims 3, 9, 12, 18, 21, 24, 27, 30, 33, 36, 39, 42, 45, 78, 89, 92

Some of these claims will be discussed individually below.

Again with respect to all of these claims, though they have different dependencies, and do not stand or fall together, the Examiner cites the same portions of the Chang reference, namely processor: col. 6, lines 21-22; computer-readable medium col. 6, line 59; and for the mapping col. 14, line 34.

Again, Applicants do not claim just any processor. Applicants claim a processor, including the medium of the prior claim and configured to use the medium to produced an XML document based on the mapping.

Chang's processor at col. 6, lines 21-22 is stated merely to store relational databases. The medium, at col. 6, line 59, was discussed in the previous group. Applicants are not finding that col. 14, line 34 is discussing a mapping to XML elements and attributes. Applicants are finding that this section of the reference discusses an indexing scheme for indexing stored XML documents. The XML elements and attributes do not appear, so far as Applicants can tell, to be produced from lists and scalars, but rather are stored in XML form.

Applicants accordingly respectfully submit that the Examiner is mischaracterizing the reference with respect to these claims.

Claim 10, 49, 64

These claims recite expressing the mapping in constructs of a mapping language. Against this recitation, the Examiner cites col. 14, line 34 et seq. Applicants respectfully submit that the Examiner mischaracterizes the reference. This section of the reference relates to indexing existing XML documents, not to a mapping language that maps lists and scalars into XML elements and attributes.

More information about the general definition of the term “language” is included in the appendix from the online encyclopedia “wikipedia.org.” According to this definition, “a language is a system of symbols, generally known as lexemes and the rules by which they are manipulated.” Applicants do not believe that a mere indexing teaches or suggests a language. In fact, the indexing scheme of the reference appears to chop up XML documents into a table, rather than being a use of constructs of a mapping language.

Applicants accordingly respectfully submit that the Examiner has failed to make a *prima facie* case against these claims.

Chang: col. 9, line 11 (claims 90, 94, 96)

This portion of the reference is cited by the Examiner in several places as allegedly showing an annotated the DTD. Applicants respectfully submit that the Examiner has misconstrued the reference. As the Examiner states, DTDid is an integer value identifying a

document type definition. This is the name of the DTD, not the contents of the DTD.

Claim 90 recites “inserting the constructs into a DTD to create an annotated DTD,” merely creating an identifier for a DTD fails to teach or suggest the limitations of this claim.

Claims 94 and 96 have similar limitations.

Chang: col. 15, lines 50-67 (claims 76, 79, 80)

These claims recite that an XML document generated from the annotated DTD is guaranteed to conform to the DTD. The DTD corresponds to multiple heterogeneous data sources.

The portion of the reference cited by the Examiner starts out “***If*** the XML documents conform to a single DTD.[emphasis added]” Please note the ***if***. Clearly from this text there is no guarantee that the XML documents will conform to the single DTD. It is only happenstance.

The Examiner says that the XML documents are multiple heterogeneous data sources – but the claim says that the XML elements and attributes are supposed to be the target, not the source. Moreover, the XML documents in the reference are all from a single database, not multiple data sources.

Applicants accordingly respectfully submit that the Examiner has failed to make a *prima facie* case against these claims and has mischaracterized the reference.

Claims 16, 51, 66

Against these claims the Examiner cites figs. 11 & 12 and col. 15, line 56 *et seq.* of

Chang. First, it should be noted that the text cited by the Examiner does not correspond with the figures cited.

Second Figures 11 and 12 of the reference, described at col. 16, line 56 et seq. relate to indexing XML documents based on tagged words in those documents (Fig. 11) and by structure of the documents (Fig. 12.) They fail to teach or suggest constructs in a DTD, much less constructs that include either a value specification or a binding specification per this group of claims.

Third, the text cited by the Examiner, referring to figures 9 and 10, relates to a structure index for an XML document database, not to contents of constructs in a DTD.

Applicants accordingly respectfully submit that the Examiner has failed to make a *prima facie* case against these claims and has mischaracterized the reference.

Claim 19, 52, 67

These claims recite that a construct comprises a parameter. This is a dependent claim, so reference must be had to the prior claims, which explain that the construct is one that is inserted into a DTD.

Against this recitation, the Examiner cites col. 20, line 63 of Chang. The undersigned has reviewed this portion of Chang and respectfully submits that the Examiner mischaracterizes it. This portion of the reference talks about parameters of a function “xml/FromFile.” Applicants see no teaching or suggestion that these parameters have anything to do with constructs of a mapping language that are inserted into a DTD. Instead, the function appears to be in the XML extender (100, Fig. 2), which appears to be an entire suite of programs.

The Examiner further cites col. 22, lines 18-57 as relating to generation of an XML element. Applicants have reviewed this portion of the reference and respectfully submit that the Examiner mischaracterizes it. Applicants find that this portion relates again to retrieving information about of an XML document using “conditional select” rather than generating XML elements and attributes.

The Examiner further cites col. 23, lines 5-51. This section does appear to relate to updating an XML document, but again it appears to use XML extender (100, Fig. 2) rather than constructs of a mapping language inserted into a DTD, per claim 19.

Applicants accordingly respectfully submit that the Examiner has failed to make a *prima facie* case against these claims and misconstrues the reference.

Claim 21

The Examiner refers to col. 22, lines 18-57. This section relates to using parameters in SQL – a query language – to search a database of XML documents. It fails to teach or suggest passing a value to a parameter that is in a construct in a DTD. Applicants accordingly respectfully submit that the Examiner has failed to make a *prima facie* case against these claims and misconstrues the reference.

Claims 25, 54, 69

Against these claims the Examiner cites col. 5, line 50 et seq. This section relates to a DTD, but not a DTD construct as defined in the parent claims of this claim. *A fortiori* this section of the reference cannot teach or suggest that data objects or formulae are associated with

such a construct. Applicants accordingly respectfully submit that the Examiner has failed to make a *prima facie* case against these claims and misconstrues the reference.

Claim 27

The associating operation of this claim is related to the independent claim 25, where associating operation is further defined as associating lists or formulas with DTD constructs having a repetition symbol.

The Examiner purports to find the “associating operation” of this claim at col. 8, lines 21-42 of Chang. Applicants have reviewed this section and see no teaching or suggestion of associating anything with a DTD construct that has a repetition symbol. Applicants accordingly respectfully submit that the Examiner has failed to make a *prima facie* case against these claims and misconstrues the reference.

Claims 28, 55, 70

Against the recitations of these claims, the Examiner cites Chang’s col. 5, lines 17-48. This section shows an example of an XML document. The section is also followed by a section that shows document type definitions for the XML document. These type definitions include a #PCDATA’s, element lists, and attribute lists.

Applicants do not claim the concept of XML documents or the type definitions per se. These are known from the art. What Applicants claim is associating one or more lists of data objects or formulas producing data objects with DTD constructs, such as shown in the example on page 21.

Applicants accordingly respectfully submit that the portion of Chang cited by the Examiner fails to teach or suggest this group of claims.

Claims 34, 57, 72

These claims recite first associating one or more lists of data objects or formulas producing data objects with a DTD construct. Examiner purports to find this first associating operation at col 5, l. 50 and col. 9, line eleven of the reference. Applicants respectfully submit that the Examiner misconstrues the reference. Column 5, line 50, et seq. merely shows a DTD. Applicants do not find that it teaches or suggests associating anything with DTD constructs, as that term is defined in the present application and claims.

These claims further recite second associating least one of the lists or formulas with at least one variable name. The examiner purports to find this at column 24, lines 38-66 of the reference. Applicants respectfully submit that the Examiner misconstrues the reference. Applicants have reviewed this section of the reference and understand it to describe searching the XML database. Applicants find no teaching or suggestion of their second associating step.

Claim 37, 58, 73

These claims recite associating at least one environment with an XML element. It is to be noted that “environment” is defined in the specification at page 31, i.e. ‘a set of variable/value pairs called the “environment.”’

Against this recitation, the Examiner cites Chang, col. 8, lines 31-32. Applicants have reviewed this part of the reference, which refers to creating a table. As far as Applicants can tell,

this table stores the XML itself, not variable/value pairs. Applicants accordingly respectfully submit that the Examiner has therefore failed to make a *prima facie* case against these claims.

Claims 40, 59, 74

These claims recite details of the environment. These claims depend from those in the previous group. Against these claims, the Examiner cites col. 15, line 50 through col. 17, line 64.

Applicants are totally unable to discern how this large amount of text may be related to an “environment” as defined by Applicants. Applicants accordingly respectfully submit that the Examiner has failed to make a *prima facie* case against these claims.

Claims 43, 60, 75

Against these claims, the Examiner cites col. 15, line 25 through col. 16, line 24. This section is entitled “Structure Index for the XML Extender” and describes something called a “structure search” of the XML documents in a database. Applicants see no relationship between this section of the reference and this group of claims.

Applicants have in the first place not found any teaching or suggestion of a mapping, as defined and discussed with respect to claim 1 above. *A fortiori* Applicants find no teaching or suggestion of what might be in such a mapping, such as a specification of a parameter for receiving a value up on generation of an XML document, per this group of claims. It appears that XML documents are being searched in this part of the reference. Applicants do not understand that XML elements and attributes are being mapped to.

The claim further references the “environment” as discussed with respect to other claims herein, and which Applicants have previously discussed as not being taught or suggested by the reference.

Applicants accordingly respectfully submit that the Examiner has not made a *prima facie* case against these claims.

Claims 84, 85, 86

Against the recitations of these claims, the Examiner cites Chang at col. 16, lines 1-22. Again, this section of the reference relates to a structure search of the database of XML documents. Applicants find no teaching or suggestion here that a user can specify anything with respect to a mapping from lists and scalars to XML elements and attributes. Applicants accordingly respectfully submit that the Examiner has not made a *prima facie* case against these claims and misconstrues the reference.

Claims 87, 93, 95

These claims recite that there are at least two data sources and the sources are of different types. Against this recitation, the Examiner cites elements 500 and 300 in Fig. 3. However, element 500 is the source and element 300 is the target in this figure. They are not both sources. Accordingly, Applicants respectfully submit that the Examiner has not made a *prima facie* case against these claims.

Decision on petition/Evidence appendices/date of invention

Declarations under rule 131 have been filed in this application. The history here is very confusing.

Originally, the declarations of inventor Shyh-kwei Chen and his manager were submitted in 2004 with respect to another reference. Inventor Chen explained in his declaration that Lo was out of the country, no longer working for the assignee, and therefore difficult to contact.

Applicants understood that the first two declarations had been entered as of right when a new non-final office action issued in April of 2005. Applicants continue to believe that even though the Examiner later complained of the lack of declaration of other inventor, these two declarations were entered as of right.

The Examiner raised objections to lack of declaration by the other inventor, Lo, erroneously citing the provisions of 37 CFR 1.47, as requiring both inventors to sign, when in fact that provision only applies to an original oath or declaration under rule 37 CFR 1.63. This erroneous citation of the law by the Examiner was confusing and misleading. Accordingly, Applicants, knowing the second inventor to be in Asia and difficult to contact, contested the requirement that the second declaration be submitted.

Applicants understand Rule 37 CFR 1.131 to be liberal with respect to who may file declarations establishing date of invention. The language says a declaration “*may*” be filed by certain identified parties, one of which is “the inventor.” The language does **not** say a declaration **must** be filed by all inventors.

After final, Applicants did submit the declaration of the second inventor. Even then, Applicants were forced to take an extension, because it was difficult to communicate with the

inventor in Taiwan and he had changed addresses since the last known address. The Examiner did not enter the second declaration.

Applicants petitioned from that refusal, and an adverse decision on appeal issued, based on MPEP 715.04, which states that both inventors must sign. This section of the MPEP cites no legal basis for both inventors having to sign, in view of the permissive language of the rule. The MPEP is not law and was certainly not promptly cited to the undersigned.

The position taken by the MPEP seems odd. The question here is “What was the date of invention?” The procedural issue of whether the second inventor signed or not is not dispositive of the facts of the date of invention. In this case, one of the inventors and a manager, working for the assignee (owner), of the application filed declarations with extensive supporting evidence. This is sufficient to demonstrate the facts. The requirement that the second inventor, Lo, file a declaration under rule 131 is not necessary to any finding of the truth here.

Applicants recognize that in light of various rule changes, the PTO may take the position that Applicants cannot submit the evidence necessary here. Applicants respectfully submit that the Board should be flexible about this. The point of the patent law is not to enforce voluminous rules, but to secure to inventors the benefit of their inventions. Even though the procedures may not allow Applicants to attach the declarations as appendices to this brief, the Board can very easily view them on line through the PTO’s automated databases. If necessary, the Board may treat this section of the brief as a petition to suspend the rules under rule 37 CFR 1.183.²

² Any fees other than the issue fee may be charged to deposit account 50-0510.

Applicants respectfully submit that the Board should find that Chang is not prior art, because:

1. The issue here is not procedural, but substantive – when was the date of invention? – and determination of the substantive issue is what is of concern here. The declarations of first submitted, by Chen and his manager, adequately demonstrate that Chang is not prior art.
2. The Examiner delayed in requiring the declaration of the second inventor, ignored the difficulty in contacting the inventor who was in Asia, and misleadingly cited the wrong regulation for the requirement of the additional declaration – and should therefore be equitably estopped from refusing to consider the additional declaration.
3. The Board should act to correct the erroneous interpretation of law in MPEP 715.04.

Applicants also respectfully submit that the declarations do establish that the invention here was conceived in the summer of 1998 and that the inventors began working on reducing it to practice diligently beginning at the latest in September of 1998. The date of filing of Chang is June of 1999.

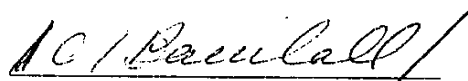
There is a provisional application in Chang cited from October of 1998, which is after the date on which Applicants have established that they started diligently working to reduce this to practice. In any case, that provisional application has not been made of record and has not been applied against the claims – so it is not clear what it teaches or suggests. The date of the provisional application from Chang is therefore not at issue here.

Applicants accordingly respectfully submit that the rule 131 declarations do establish that Chang is not prior art.

VIII. CONCLUSION

Applicants respectfully submit that they have answered each issue raised by the Examiner and that the application is accordingly in condition for allowance. Such allowance is therefore respectfully requested.

Respectfully submitted,



By _____

Anne E. Barschall

Reg. No. 31,089

(914) 332-1019

fax 914-332-7719

November 8, 2007

CLAIMS APPENDIX

1. A computer method, comprising executing at least the following operation in at least one data processing device:

establishing a mapping from lists and scalars corresponding to at least one data source into XML elements and attributes.

2. At least one medium readable by a data processing device and embodying at least one result of the method of claim 1.

3. A data processing device comprising:

- the at least one medium according to claim 2, and
- at least one processor configured to use the at least one medium to produce an XML document based on the mapping.

4-6. (cancelled)

7. The method of claim 1, wherein the data source is a relational database.

8. At least one medium readable by a data processing device and embodying at least one result of the method of claim 7.

9. A data processing device comprising

- the at least one medium according to claim 8; and
- at least one processor configured to use the at least one medium to produce the XML document based on the mapping.

10. The method of claim 1, further comprising executing the following operation in the data processing device:

expressing the mapping in constructs of a mapping language.

11. At least one medium readable by a data processing device and embodying at least one result of the method of claim 10.

12. A data processing device comprising

- the at least one medium according to claim 11; and
- at least one processor configured to use the at least one medium to produce an XML document based on the mapping.

13-15 (cancelled).

16. The method of claim 90, wherein the constructs comprise at least one of a value specification and a binding specification.

17. At least one medium readable by a data processing device and embodying at least one result of the method of claim 16.

18. A data processing device comprising:

- the at least one medium according to claim 17; and
- at least one processor configured to use the at least one medium to produce an XML document based on the annotated DTD.

19. The method of claim 90, wherein

- at least one of the constructs comprises at least one parameter;
- the at least one of the constructs is adapted so that a value of the at least one of the parameters is determinable at a time of generation of at least one respective XML element associated with the at least one of the constructs.

20. At least one medium readable by a data processing device and embodying at least one result of the method of claim 19.

21. A data processing device comprising:

- the at least one medium according to claim 20; and
- at least one processor configured to
 - use the at least one medium to produce an XML document based on the mapping;and

- pass the value to the parameter.

22. The method of claim 90, further comprising executing the following operation in the data processing device: associating values and or formulas with the DTD.

23. At least one medium readable by a data processing device and embodying at least one result of the method of claim 22.

24. A data processing device comprising:

- the at least one medium according to claim 23; and
- at least one processor configured to
 - use the at least one medium to produce an XML document based on the mapping ;
 - and
 - perform the associating operation.

25. The method of claim 22, wherein the associating includes associating one or more lists of data objects or formulas producing data objects with each DTD construct having a repetition symbol at the end.

26. At least one medium readable by a data processing device and embodying at least one result of the method of claim 25.

27. A data processing device comprising:

- at least one medium according to claim 26; and
- at least one processor configured to
 - use the at least one medium to produce an XML document; and
 - perform the associating operation.

28. The method of claim 22, wherein the associating includes associating one or more lists of data objects or formulas producing data objects with each DTD construct which is not a #PCDATA, a choice list, or an attribute list, and does not end with a repetition symbol.

29. At least one medium readable by a data processing device and embodying at least one result of the method of claim 28.

30. A data processing device comprising:

- the at least one medium according to claim 29; and
- at least one processor configured to
 - use the at least one medium to produce an XML document based on the mapping; and
 - perform the associating operation.

31. The method of claim 22, wherein associating includes associating a value or formula producing a value with each PCDATA, choice list, or attribute definition.

32. At least one medium readable by a data processing device and embodying at least one result of the method of claim 31.

33. A data processing device comprising:

- the at least one medium according to claim 32; and
- at least one processor configured to
 - use the at least one medium to produce an XML document; and
 - perform the associating operation.

34. The method of claim 22, wherein associating includes, not necessarily in the following order:

- first associating one or more lists of data objects or formulas producing data objects with a DTD construct;
- second associating at least one of the lists or formulas with at least one variable name; and
- using the variable name as a parameter in at least one other formula.

35. At least one medium readable by a data processing device and embodying at least one result of the method of claim 34.

36. A data processing device comprising:

- the at least one medium according to claim 35; and
- at least one processor configured to

- use the at least one medium to produce an XML document; and
- perform the associating operation and included operations.

37. The method of claim 1, further comprising executing the following operation in the data processing device: associating at least one respective environment with a respective XML element to be generated.

38. At least one medium readable by a data processing device and embodying at least one result of the method of claim 37.

39. A data processing device comprising:

- the at least one medium according to claim 38; and
- at least one processor configured to
 - use the at least one medium to produce an XML document; and
 - perform the associating operation.

40. The method of claim 37, wherein the at least one environment comprises

- information from a parent XML element of the respective XML element; and
- information from a binding specification of a DTD construct associated with the respective XML element.

41. At least one medium readable by a data processing device and embodying at least one result of the method of claim 40.

42. A data processing device comprising:

- the at least one medium according to claim 41; and
- at least one processor configured to
 - use the at least one medium to produce an XML document; and
 - perform the associating operation.

43. The method of claim 37, wherein

- the mapping includes at least one respective specification corresponding to at least one respective XML element;
- the specification comprises at least one parameter for receiving a value upon generation of an XML document; and
- the method further comprises, upon generation of an XML document, sending the at least one parameter a value according to at least one variable/value pair in the at least one respective environment.

44. At least one medium readable by a data processing device and embodying at least one result of the method of claim 43.

45. A data processing device comprising:

- the at least one medium according to claim 44; and
- at least one processor configured to
 - use the at least one medium to produce an XML document; and
 - perform the associating and sending operations.

46. At least one medium readable by at least one data processing device and embodying software adapted to perform operations comprising:

establishing a mapping from lists and scalars corresponding to at least one data source into XML elements and attributes.

47. (canceled)

48. The at least one medium of claim 46, wherein the data source is a relational database.

49. The at least one medium of claim 46, wherein the operations further comprise:
expressing the mapping in constructs of a mapping language.

50. (canceled)

51. The at least one medium of claim 94, wherein the constructs comprise at least one of a value specification and a binding specification.

52. The at least one medium of claim 94, wherein

- at least one of the constructs comprises at least one parameter; and
- the at least one of the constructs is adapted so that a value of the at least one of the parameters is determinable at a time of generation of at least one respective XML element associated with the at least one of the constructs.

53. The at least one medium of claim 94, wherein the operations further comprise associating values and or formulas with the annotated DTD.

54. The at least one medium of claim 53, wherein the associating includes associating one or more lists of data objects or formulas producing data objects with each DTD construct having a repetition symbol at the end.

55. The at least one medium of claim 54, wherein the associating includes associating one or more lists of data objects or formulas producing data objects with each DTD construct which is not a #PCDATA, a choice list, or an attribute list, and does not end with a repetition symbol.

56. The at least one medium of claim 54, wherein associating includes associating a value or formula producing a value with each PCDATA, choice list, or attribute definition.

57. The at least one medium of claim 54, wherein associating includes, not necessarily in the following order:

- first associating one or more lists of data objects or formulas producing data objects with a DTD construct;
- second associating at least one of the lists or formulas with at least one variable name; and
- using the variable name as a parameter in at least one other formula.

58. The at least one medium of claim 46, wherein the operations further comprise associating at least one respective environment with a respective XML element to be generated.

59. The at least one medium of claim 58, wherein the at least one environment comprises

- information from a parent XML element of the respective XML element; and
- information from a binding specification of a DTD construct associated with the respective XML element.

60. The at least one medium of claim 58, wherein

- the mapping includes at least one respective specification corresponding to at least one respective XML element;
- the specification comprises at least one parameter for receiving a value upon generation of an XML document; and
- the method further comprises, upon generation of an XML document, sending the at least one parameter a value according to at least one variable/value pair in the at least one respective environment.

61. At least one data processing device comprising:

- means for receiving data from at least one data source;
- at least one processor adapted to perform operations comprising:
 - establishing a mapping from lists and scalars corresponding to the data into XML elements and attributes.

.

62. (canceled)

63. The at least data processing device of claim 61, wherein

- the at least one data source comprises at least two data sources, and the data sources are of different types; and
- the data sources are relational databases.

64. The at least one data processing device of claim 61, wherein the operations further comprise: expressing the mapping in constructs of a mapping language.

65. (canceled)

66. The at least one data processing device of claim 64, wherein the constructs comprise at least one of a value specification and a binding specification.

67. The at least one data processing device of claim 64, wherein

- at least one of the constructs comprises at least one parameter; and
- the at least one of the constructs is adapted so that a value of the at least one of the parameters is determinable at a time of generation of at least one respective XML element associated with the at least one of the constructs.

68. The at least one data processing device of claim 96, wherein the operations further comprise associating values and or formulas with the annotated DTD.

69. The at least one data processing device of claim 68, wherein the associating includes associating one or more lists of data objects or formulas producing data objects with each DTD construct having a repetition symbol at the end.

70. The at least one data processing device of claim 68, wherein the associating includes associating one or more lists of data objects or formulas producing data objects with each DTD construct which is not a #PCDATA, a choice list, or an attribute list, and does not end with a repetition symbol.

71. The at least one data processing device of claim 68, wherein the associating includes associating a value or formula producing a value with each PCDATA, choice list, or attribute definition.

72. The at least one data processing device of claim 68, wherein the associating includes, not necessarily in the following order:

- first associating one or more lists of data objects or formulas producing data objects with a DTD construct;
- second associating at least one of the lists or formulas with at least one variable name; and
- using the variable name as a parameter in at least one other formula.

73. The at least one data processing device of claim 61, wherein the operations further comprise associating at least one respective environment with a respective XML element to be generated.

74. The at least one data processing device of claim 73, wherein the at least one environment comprises

- information from a parent XML element of the respective XML element; and
- information from a binding specification of a DTD construct associated with the respective XML element.

75. The at least one data processing device of claim 73, wherein

- the mapping includes at least one respective specification corresponding to at least one respective XML element;
- the specification comprises at least one parameter for receiving a value upon generation of an XML document; and

- the method further comprises, upon generation of the XML document, sending the at least one parameter a value according to at least one variable/value pair in the at least one respective environment

76. The method of claim 1, wherein

- the at least one data source comprises multiple heterogeneous data sources; and
- the method further comprises
- using a pre-established DTD corresponding to the multiple heterogeneous data sources; and
- based on the DTD and the multiple heterogeneous data sources, adding annotations to the DTD to create an annotated DTD, such that an XML document generated from the annotated DTD is guaranteed to conform to the DTD.

77. At least one medium readable by a data processing device and embodying at least one result of the method of claim 76.

78. A data processing device comprising:

- the at least one medium according to claim 77; and
- at least one processor configured to use the at least one medium to produce the XML document based on the mapping.

79. The medium of claim 46, wherein

- at least one data source comprises multiple heterogeneous data sources; and

- the operations further comprise
- using a pre-established DTD corresponding to the multiple heterogeneous data sources ; and
- based on the DTD and the multiple heterogeneous data sources, adding annotations to the DTD to create an annotated DTD, such that an XML document generated from the annotated DTD is guaranteed to conform to the DTD.

80. The data processing device of claim 61, wherein

- the at least one data source comprises multiple heterogenous data sources; and
- the operations further comprise
- using a pre-established DTD corresponding to the multiple heterogeneous data sources; and
- based on the DTD and the multiple heterogeneous data sources, adding annotations to the DTD to create an annotated DTD, such that an XML document generated from the annotated DTD is guaranteed to conform to the DTD.

81-83 (cancelled)

84. The method of claim 1, wherein the mapping is responsive to a user mapping specification.

85. The medium of claim 46, wherein the mapping is responsive to a user mapping specification.

86. The data processing device of claim 61, wherein the mapping is responsive to a user mapping specification.

87. The method of claim 1, wherein the at least one data source comprises at least two data sources, and the data sources are of different types.

88. At least one medium readable by a data processor and embodying at least one result of the method of claim 87.

89. A data processing device comprising:

- the at least one medium according to claim 88; and
- at least one processor configured to use the at least one medium to produce an XML document based on the mapping.

90. The method of claim 10, further comprising executing the following operation in the data processing device: inserting the constructs into a DTD to create an annotated DTD.

91. At least one medium readable by a data processing device and embodying at least one result of the method of claim 90.

92. A data processing device comprising:

- the at least one medium according to claim 91; and
- at least one processor configured to

- use the at least one medium to produce an XML document based on the mapping;
and
- perform the inserting operation.

93. The at least one medium of claim 46, wherein the at least one data source comprises at least two data sources, and the data sources are of different types.

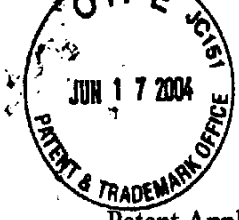
94. The at least one medium of claim 46, further comprising executing the following operation in the data processing device: inserting the constructs into a DTD to create an annotated DTD.

95. The at least one data processing device of claim 61, wherein the at least one data source comprises at least two data sources, and the data sources are of different types.

96. The at least one data processing device of claim 64, further comprising executing the following operation in the data processing device: inserting the constructs into a DTD to create an annotated DTD.

EVIDENCE APPENDIX

Enclosed please find declarations of inventor Chen and his manager under rule 131. Applicants respectfully submit that these declarations were entered into the record as of right when the Examiner issued a new non-final action April 20, 2005, whether the Examiner said so or not.



IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Patent Application Ser. No.: 09/466,627

Group Art Unit: 2176

Filing Date: 12/17/1999

Examiner: M. NGUYEN

Attorney Docket Number YO999-429

Inventor Name(s): LO ET AL.

Title: METHOD AND APPARATUS FOR CONVERTING BETWEEN DATA SETS AND
XML DOCUMENTS

Commissioner for Patents
P.O. Box 1450
Alexandria VA 223131-1450

DECLARATION OF JEN-YAO CHUNG, PH.D.

Sir:

I, Jen-Yao Chung, Ph.D., hereby declare as follows:

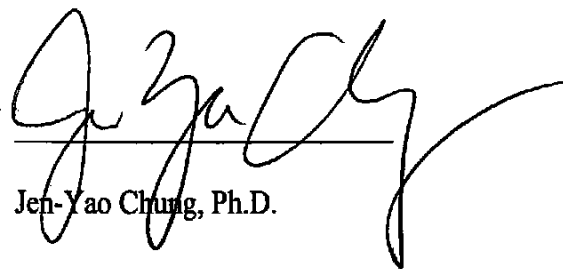
1. During the years 1998 and 1999, I was manager to Shyh-Kwei Chen and responsible for supervising his project relating to converting database information to XML. I was also familiar with the work of his co-inventor Ming-Ling Lo.
2. Attached as Exhibit A is a copy of an e-mail from Denise Dyko to Kevin Leahy and Michael Swanson dated September 17th, 1998. This e-mail mentions to store and retrieve XML data in current enterprise data formats, and to enable rapid exploitation of existing data in e-commerce XML applications while continuing to support existing applications against that data, relating to this project. This e-mail is a printout from my personal computer and is a business record that I maintain. I therefore trust its accuracy.

3. This e-mail evidences the proposed project which matured into the patent application identified above. Based on this e-mail, I believe that Shyh-Kwei Chen and Ming-Ling Lo conceived of their invention, namely the idea of establishing a mapping from lists and scalars corresponding to at least one data source into XML elements and attributes, during the summer of 1998.
4. Attached as Exhibit B, hereto are three e-mails from Denise Dyko to myself dated September 10th, 1998, from myself to Denise Dyko and Chungti Liang dated September 25th, 1998, and from Denise Dyko to Don Ault, John Thompson, and Eric Porter dated October 13th, 1998. These emails are business records I maintain in my computer and I therefore trust their accuracy. These emails describe the basis for a universal data management architecture, which lead to the funding for the XML Access Server OS/390 project. Confidential information not relating to the present application has been blacked out; however, the information relevant to establishing dates for this application is still visible.
5. Attached as Exhibit C, hereto is an e-mail from Denise Dyko to Gerry Meyer dated November 30th, 1998. This email is a business record I maintain in my computer and I therefore trust its accuracy. This email explains that Shyh-Kwei Chen and Ming-Ling Lo were projected to work full time on the project (2PY) that matured into the present application during all of 1999. Confidential information not relating to the present application has been blacked out; however, the information relevant to establishing dates for this application is still visible.
6. Attached as Exhibit D, hereto are five e-mails from Denise Dyko to Ming-Ling Lo dated December 3rd, 1998, from Denise Dyko to George Zagelow dated December 4th, 1998,

from Denise Dyko to Gerry Meyer dated December 4th, 1998, from Denise Dyko to David Fallside dated December 9th, 1998, and from myself to Denise Dyko dated December 10th, 1998. These emails are business records I maintain in my computer and I therefore trust their accuracy. These emails mention that Shyh-Kwei Chen and Ming-Ling Lo were to present the newly funded XML 390 Access Server project to the XML summit that was held on December 15th and 16th, 1998.

7. Attached as Exhibit E, hereto is a set of presentation slides dated December 15th, 1998. These presentation slides are business records I maintain in my computer and I therefore trust its accuracy. The presentation outlines that Shyh-Kwei Chen and Ming-Ling Lo were to work full time during 1999 on the project that matured into the patent application identified above.
8. Based on the documents identified above, my recollection is refreshed regarding the events of 1998 and 1999. I therefore remember that Shyh-Kwei Chen and Ming-Ling Lo
 - conceived of their invention of the idea of establishing a mapping from lists and scalars corresponding to at least one data source into XML elements and attributes during the summer of 1998;
 - worked at least part time during the fall of 1998 to reduce this invention to practice;
 - worked full time reducing this invention to practice during the entire year of 1999.

7. I further declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issuing thereon.

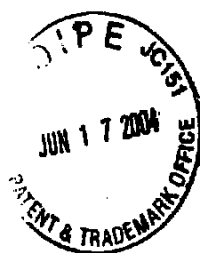


Jen-Yao Chung, Ph.D.

June 10, 2004

Date

Exhibit A



Jen-Yao Chung/Watson/IBM

05/27/2004 04:16 PM

This document expires on

05/27/2103

To ShyhKwei Chen/Watson/IBM@IBMUS

cc

bcc

Subject Re: Research Project Funding for XML in OS/390

Denise Y. Dyko 09/17/98 03:33 PM

Denise Y. Dyko 09/17/98 03:33 PM

To: Kevin Leahy/Poughkeepsie/IBM@IBMUS, Michael Swanson/Poughkeepsie/IBM@IBMUS
cc: Jen-Yao Chung/Watson/IBM@ibmus, Denise Gorski/Poughkeepsie/IBM@IBMUS, Michael
Oliver/Poughkeepsie/IBM@IBMUS, Bob Schloss/Watson/IBM@IBMUS, ShyhKwei
Chen/Watson/IBM@IBMUS, Nancy P Riggs/Watson/IBM@IBMUS, Thomas
Rozmus/Poughkeepsie/IBM@IBMUS, Douglas Archer/Poughkeepsie/IBM@ibmus, Chungti
Liang/Poughkeepsie/IBM@IBMUS, David Fallside/Santa Teresa/IBM@ibmus, Susan Malaika/Santa
Teresa/IBM@IBMUS, Mike Baskey/Poughkeepsie/IBM@IBMUS
From: Denise Y. Dyko/Poughkeepsie/IBM @ IBMUS
Subject: Research Project Funding for XML in OS/390

I met with Jen-Yao Chung, Bob Schloss, and ShyhKwei Chen yesterday in Hawthorne to review what research can best contribute to OS/390's strategy for XML. (Kevin, I'm the focal point for OS/390 XML strategy.) Our discussion resulted in the following project description:

In addition to storing documents in repositories/databases in XML format, there is a need to store and retrieve XML data in current enterprise data formats, to enable rapid exploitation of existing data in e-commerce XML applications while continuing to support existing applications against that data. This research project will propose techniques and algorithms to manage the automatic composing of XML documents from existing enterprise data, and conversion back to proprietary formats from XML. The project will include effective ways to prepare XML documents from multiple enterprise data sources using XML name spaces, and algorithms to combine/merge DTDs. Initial deliverables will include articulation of the OS/390 XML strategy and its validation via proof of concept. The OS/390 division, with research, will also undertake validation with customers and ISVs.

This project definition is aimed at rapid execution of the corporate XML strategy, which sees the value of XML as its enablement of volume growth in e-business, to the point where NT and UNIX platforms cannot scale rapidly enough to compete with OS/390.

I need your help to get this project included in the 390 division's funding of research projects. Would you like me to set up a meeting to review this project definition and OS/390's role in XML in general? With whom should I work to ensure that this project is prioritized with the other work being funded in research?

Cheers, Denise

Lotus Notes Address: Denise Y. Dyko/Poughkeepsie/IBM@IBMUS

VM Address: KGNVMC.DYKO@VM

Internet Address: DYD@VNET.IBM.COM

Phone: (914)435-6903, t/l 8-295-6903



Exhibit B

Jen-Yao Chung/Watson/IBM
05/27/2004 04:17 PM
This document expires on
05/27/2103

To ShyhKwei Chen/Watson/IBM@IBMUS
cc
bcc

Subject Re: XML project proposal and Meeting on Sept. 16, 11am

Denise Y. Dyko 09/10/98 03:50 PM

Denise Y. Dyko 09/10/98 03:50 PM

To: Jen-Yao Chung/Watson/IBM@ibmus
cc: ShyhKwei Chen/Watson/IBM@IBMUS, Michael Swanson/Poughkeepsie/IBM@IBMUS, Thomas Rozmus/Poughkeepsie/IBM@IBMUS
From: Denise Y. Dyko/Poughkeepsie/IBM @ IBMUS
Subject: Re: XML project proposal and Meeting on Sept. 16, 11am

Let's start at 1:00, in your office. The project description is excellent. Thank you.

Mike, Jen-Yao feels the project description is the basis for a universal data management architecture. (We're both very enthused about the possibilities.) Is there anyone else you feel should attend this meeting? We'll be identifying specific research activities to support this project.


Cheers, Denise

Lotus Notes Address: Denise Y. Dyko/Poughkeepsie/IBM@IBMUS

VM Address: KGNVMC.DYKO@VM

Internet Address: DYD@VNET.IBM.COM

Phone: (914)435-6903, t/l 8-295-6903



Jen-Yao Chung
09/10/98 03:23 PM

To: Denise Y. Dyko/Poughkeepsie/IBM@IBMUS
cc: ShyhKwei Chen/Watson/IBM@IBMUS
From: Jen-Yao Chung/Watson/IBM @ IBMUS
Subject: XML project proposal and Meeting on Sept. 16, 11am

Hi Denise, can you come down on Sept. 16. I'm free after 11am. We want meet from 11am or 1pm. Please let me know your availability. Thanks. -chung

Project: Enterprise Data Management Using XML

- Instead of storing documents to the repositories/databases in XML format, there is still a need for storing enterprise data in current format due to large volume and business issues. We are proposing techniques and algorithms to manage the automatic composing of XML documents from existing enterprise data, and converting back to proprietary formats from XML - We also look for effective ways to prepare XML documents from multiple enterprise data sources using XML name space, and DTD combining/merging algorithms.

Jen-Yao Chung/Watson/IBM

05/27/2004 04:15 PM

This document expires on

05/27/2103

To ShyhKwei Chen/Watson/IBM@IBMUS

cc

bcc

Subject Re: *IBM Confidential: Enterprise Data Management Using XML project detail description

Jen-Yao Chung

Jen-Yao Chung

09/25/98 01:07 PM

To: Denise Y. Dyko/Poughkeepsie/IBM@IBMUS, Chungli Liang/Poughkeepsie/IBM@IBMUS
cc: ShyhKwei Chen/Watson/IBM@IBMUS, Anant Jhingran/Watson/IBM@IBMUS
From: Jen-Yao Chung/Watson/IBM @ IBMUS
Subject: *IBM Confidential: Enterprise Data Management Using XML project detail description []

Project Name: Enterprise Data Management Using XML

Project Objective:

In addition to storing documents in repositories/databases in XML format, there is a need to store and retrieve XML data in current enterprise data formats, to enable rapid exploitation of existing data in e-commerce XML applications while continuing to support existing applications against that data.

Previous Work:

XML activities

- . Proposed transparent XML/EDI generic translation/transportation/rendering schemes, and generated a proof-of-concept prototype
- . Produced document turnaround logic
- . Filed two patents on dynamic business process automation system using XML, and automatic XML display script/style sheet and DTD generation
- Net.Data OS390 delivery
- . Included DB2 connections, national language support, and numerous 'backend scripts' integration such as REXX, PERL, and compiled-C code etc..
- . took part in Beta customer support program

Work Items:

This research project will propose techniques and algorithms to manage the automatic composing of XML documents from existing enterprise data, and conversion back to proprietary formats from XML. The project will include effective ways to prepare XML documents from multiple enterprise data sources using XML name spaces, and algorithms to combine/merge DTDs.

Potential Deliverables:

Initial deliverables will include articulation of the OS/390 XML strategy, proof-of-concept validation tools/programs, and potential patent filing and paper publication. [REDACTED]

Jen-Yao Chung/Watson/IBM

05/27/2004 04:14 PM

This document expires on

05/27/2103

To ShyhKwei Chen/Watson/IBM@IBMUS

cc

bcc

Subject Re: OE access to OS/390 legacy data stores

Denise Y. Dyko 10/13/98 05:42 PM

Denise Y. Dyko 10/13/98 05:42 PM

To: Don Ault/Poughkeepsie/IBM@ibmus, John Thompson/Poughkeepsie/IBM@IBMUS, Eric Porter/Poughkeepsie/IBM@IBMUS

cc: Jen-Yao Chung/Watson/IBM@ibmus

From: Denise Y. Dyko/Poughkeepsie/IBM @ IBMUS

Subject: Re: OE access to OS/390 legacy data stores

Thank you all for your responsiveness. John, we have defined the following research project related to XML on OS/390:

In addition to storing documents in repositories/databases in XML format, there is a need to store and retrieve XML data in current enterprise data formats, to enable rapid exploitation of existing data in e-commerce XML applications while continuing to support existing applications against that data. This research project will propose techniques and algorithms to manage the automatic composing of XML documents from existing enterprise data, and conversion back to proprietary formats from XML. The project will include effective ways to prepare XML documents from multiple enterprise data sources using XML name spaces, and algorithms to combine/merge DTDs. Initial deliverables will include articulation of the OS/390 XML strategy and its validation via proof of concept. The OS/390 division, with research, will also undertake validation with customers and ISVs.

The sentence I've marked in blue was the impetus for determining how OE references MVS datasets today. I realize this project description is high-level, but the project is just now getting kicked off.

Cheers, Denise

Lotus Notes Address: Denise Y. Dyko/Poughkeepsie/IBM@IBMUS

VM Address: KGNVMC.DYKO@VM

Internet Address: DYD@VNET.IBM.COM

Phone: (914)435-6903, t/ 8-295-6903

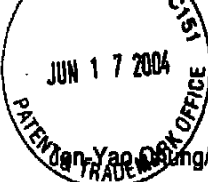


Exhibit C

Jen-Yao Chung/Watson/IBM
05/27/2004 04:14 PM
This document expires on
05/27/2103

To ShyhKwei Chen/Watson/IBM@IBMUS
cc
bcc
Subject Re: *IBM Confidential: XML Activities and Contacts for
AS/400, RS/6000, S/390

Best regards,
Jen-Yao Chung
CTO, Global Electronics Industry (GEI)
914/945-3422 (T/L 862) jychung@us.ibm.com
<http://www.research.ibm.com/people/j/jychung>
Denise Y. Dyko 11/30/98 05:44 PM

Denise Y. Dyko 11/30/98 05:44 PM

To: Gerry Meyer/Rochester/IBM@IBMUS
cc: David Boutcher/Rochester/IBM@IBMUS, De Vonna Naivar/Austin/IBM@IBMUS, Theresa
Backlund/Rochester/IBM@IBMUS, Mike Tomashek/Rochester/IBM@IBMUS, Gerry
Hackett/Austin/IBM@IBMUS, Thomas Rozmus/Poughkeepsie/IBM@IBMUS, Doug
Balog/Poughkeepsie/IBM@IBMUS, Jim Porell/Poughkeepsie/IBM@IBMUS, Jen-Yao
Chung/Watson/IBM@ibmus
From: Denise Y. Dyko/Poughkeepsie/IBM @ IBMUS
Subject: Re: *IBM Confidential: XML Activities and Contacts for AS/400, RS/6000, S/390

Hi, Gerry, we still haven't managed to connect by phone. Our XML efforts are at "workbook" level, not yet strategy document. I've attached the first draft of the OS/390 XML workbook. Our key focus is on infrastructure and we have funded 2PY in Reserach for the following project definition (which is also included in the Workbook):

RESEARCH PROJECT DEFINITION:

In addition to storing documents in repositories/databases in XML format, there is a need to store and retrieve XML data in current enterprise data formats, to enable rapid exploitation of existing data in e-commerce XML applications while continuing to support existing applications against that data. This research project will propose techniques and algorithms to manage the automatic composing of XML documents from existing enterprise data, and conversion back to proprietary formats from XML. The project will include effective ways to prepare XML documents from multiple enterprise data sources using XML name spaces, and algorithms to combine/merge DTDs. Initial deliverables will include articulation of the OS/390 XML strategy and its validation via proof of concept. The OS/390 division, with research, will also undertake validation with customers and ISVs.

Other work will include support of XML by individual products/components (e.g., DB2 as an XML repository) and exploitation of XML by individual products/components in their own implementation (e.g., to replace existing proprietary data formats). But we have no overall OS/390 product plan yet --and certainly not one that is funded. We did begin testing of the XML parser on OS/390 and ran into EBCDIC/ASCII problems; I will be interested to hear about your experiences with the XML parser on AS/400.

Hope to talk to you soon.

Cheers, Denise

Lotus Notes Address: Denise Y. Dyko/Poughkeepsie/IBM@IBMUS

VM Address: KGNVMC.DYKO@VM

Internet Address: DYD@VNET.IBM.COM

Phone: (914)435-6903, t/l 8-295-6903

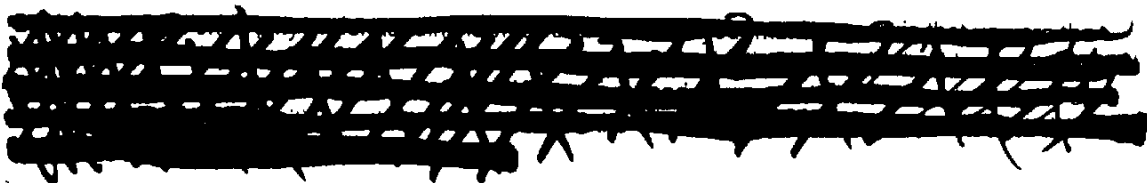


Gerry Meyer




11/23/98 10:53

To: Denise Y. Dyko/Poughkeepsie/IBM@IBMUS, De Vonna Naivar/Austin/IBM@IBMUS
cc: David Boutcher/Rochester/IBM@IBMUS, Jim Herring/Rochester/IBM@IBMUS, Theresa
Backlund/Rochester/IBM@IBMUS, Mike Tomashek/Rochester/IBM@IBMUS, Gerry
Hackett/Austin/IBM@IBMUS, Thomas Rozmus/Poughkeepsie/IBM@IBMUS, Doug
Balog/Poughkeepsie/IBM@IBMUS
From: Gerry Meyer/Rochester/IBM@IBMUS
Subject: IBM Confidential: XML Activities and Contacts for AS/400, RS/6000, S/390



Please complete the attached Freelance chart for your system showing your XML contacts and activities
Please add any additional categories, activities, etc. Give me or Dave Boutcher a call for any questions

Gerry Meyer
Senior Technical Staff Member
AS/400 System Software Development
Internet ID: gerrym@us.ibm.com
Ext 3-7266

XML Activities and Contacts.PRZ has been deleted (was saved in repository My Attachments
Repository -> ) from this note on 14 August 2000 by Jen-Yao Chung


390xmlwkbk.lwp has been deleted (was already in repository My Attachments Repository -> ) from
this note on 14 August 2000 by Jen-Yao Chung



Exhibit 1

Jen-Yao Chung/Watson/IBM
05/27/2004 04:13 PM
This document expires on
05/27/2103

To ShyhKwei Chen/Watson/IBM@IBMUS
cc
bcc
Subject Re: XML 390 foils


Denise Y. Dyko 12/03/98 06:10 PM

Denise Y. Dyko 12/03/98 06:10 PM

To: Ming-Ling Lo/Watson/IBM@IBMUS
cc: Jen-Yao Chung/Watson/IBM@ibmus, ShyhKwei Chen/Watson/IBM@IBMUS, George Zagelow/Santa Teresa/IBM@ibmus
From: Denise Y. Dyko/Poughkeepsie/IBM @ IBMUS
Subject: Re: XML 390 foils

Thank, Ming-Ling. This looks wonderful; I look forward to tomorrow's call. Would one of you be willing to present at the XML summit? (George, I think we have a presenter.)

Cheers, Denise
Lotus Notes Address: Denise Y. Dyko/Poughkeepsie/IBM@IBMUS
VM Address: KGNVMC.DYKO@VM
Internet Address: DYD@VNET.IBM.COM
Phone: (914)435-6903, t/l 8-295-6903



Ming-Ling Lo
12/03/98 05:56 PM

To: Denise Y. Dyko/Poughkeepsie/IBM@IBMUS
cc: Jen-Yao Chung/Watson/IBM@IBMUS, ShyhKwei Chen/Watson/IBM@IBMUS
From: Ming-Ling Lo/Watson/IBM @ IBMUS
Subject: XML 390 foils

Hi, Denise:
I am also involved in the XML 390 project.
In the attachment please find the set of foils for tomorrow's phone conference.
Please let me know if there is any problem.
Thanks.
-Mingling

Ming-Ling Lo
Research Staff Member, IBM T.J.Watson Research Center
Phone: (914) 784-7734, t/l: 863 7734
Email: mingling@watson.ibm.com, mll@us.ibm.com

Email: mingling@watson.ibm.com, mlio@us.ibm.com

Xml390.prz has been deleted (was already in repository My Attachments Repository -> ) from this note on 14 August 2000 by Jen-Yao Chung

Jen-Yao Chung/Watson/IBM

05/27/2004 04:13 PM
This document expires on
05/27/2103

To ShyhKwei Chen/Watson/IBM@IBMUS

cc

bcc

Subject Re: XML 390 Access Server presentation at the XML summit

Denise Y. Dyko 12/04/98 09:48 AM

Denise Y. Dyko 12/04/98 09:48 AM

To: George Zagelow/Santa Teresa/IBM@ibmus
cc: Jen-Yao Chung/Watson/IBM@ibmus, ShyhKwei Chen/Watson/IBM@IBMUS, Ming-Ling Lo/Watson/IBM@IBMUS
From: Denise Y. Dyko/Poughkeepsie/IBM @ IBMUS
Subject: XML 390 Access Server presentation at the XML summit

Hi, George, Ming-Ling Lo or Shyh-Kwei Chen can present at the XML Summit on "XML Access Server for IBM 390." A draft of the presentation is in the attached note (which I also sent you yesterday). I'd suggest 45 minutes for the presentation. The XML Access Server is the XML research project being funded by OS/390.

Cheers, Denise

Lotus Notes Address: Denise Y. Dyko/Poughkeepsie/IBM@IBMUS

VM Address: KGNVMC.DYKO@VM

Internet Address: DYD@VNET.IBM.COM

Phone: (914)435-6903, t/l 8-295-6903

Forwarded by Denise Y. Dyko/Poughkeepsie/IBM on 12/04/98 09:44 AM -----

 Ming-Ling Lo
12/03/98 05:56 PM

To: Denise Y. Dyko/Poughkeepsie/IBM@IBMUS
cc: Jen-Yao Chung/Watson/IBM@IBMUS, ShyhKwei Chen/Watson/IBM@IBMUS
From: Ming-Ling Lo/Watson/IBM @ IBMUS
Subject: XML 390 foils

Hi, Denise:

I am also involved in the XML 390 project.

In the attachment please find the set of foils for tomorrow's phone conference.

Please let me know if there is any problem.

Thanks.

-Mingling

Ming-Ling Lo
Research Staff Member, IBM T.J.Watson Research Center

Phone: (914) 784-7734, t/l: 863 7734
Email: mingling@watson.ibm.com, mlo@us.ibm.com

Xml390.prz has been deleted (was already in repository My Attachments Repository -> ) from this note on 14 August 2000 by Jen-Yao Chung

Jen-Yao Chung/Watson/IBM

05/27/2004 04:13 PM

This document expires on

05/27/2103

To ShyhKwei Chen/Watson/IBM@IBMUS

cc

bcc

Subject Re: XML 390 foils

Denise Y. Dyko 12/04/98 12:55 PM

Denise Y. Dyko 12/04/98 12:55 PM

To: Gerry Meyer/Rochester/IBM@IBMUS

cc: Jen-Yao Chung/Watson/IBM@ibmus, Ming-Ling Lo/Watson/IBM@IBMUS, ShyhKwei Chen/Watson/IBM@IBMUS

From: Denise Y. Dyko/Poughkeepsie/IBM @ IBMUS

Subject: XML 390 foils

Hi, Gerry, attached is a presentation that further defines the Research project for an XML Access Server for 390.

Ming-Ling or ShyhKwei will be presenting the XML Access Server at the XML Summit. You can contact Jen-Yao Chung (project manager) or either of them for more information.

Cheers, Denise

Lotus Notes Address: Denise Y. Dyko/Poughkeepsie/IBM@IBMUS

VM Address: KGNVMC.DYKO@VM

Internet Address: DYD@VNET.IBM.COM

Phone: (914)435-6903, t/l 8-295-6903

----- Forwarded by Denise Y. Dyko/Poughkeepsie/IBM on 12/04/98 12:09 PM -----

 Ming-Ling Lo

12/03/98 05:56 PM

To: Denise Y. Dyko/Poughkeepsie/IBM@IBMUS

cc: Jen-Yao Chung/Watson/IBM@IBMUS, ShyhKwei Chen/Watson/IBM@IBMUS

From: Ming-Ling Lo/Watson/IBM @ IBMUS

Subject: XML 390 foils

Hi, Denise:

I am also involved in the XML 390 project.

In the attachment please find the set of foils for tomorrow's phone conference

Please let me know if there is any problem.

Thanks.

-Mingling

Ming-Ling Lo

Research Staff Member, IBM T.J.Watson Research Center

Phone: (914) 784-7734, t/l: 863 7734

Xmi390.prz has been deleted (was already in repository My Attachments Repository -> ) from this note on 14 August 2000 by Jen-Yao Chung

Jen-Yao Chung/Watson/IBM

05/27/2004 04:12 PM

This document expires on

05/27/2103

To ShyhKwei Chen/Watson/IBM@IBMUS

cc

bcc

Subject Re: XML 390 Access Server presentation at the XML summit

Denise Y. Dyko 12/09/98 02:29 PM

Denise Y. Dyko 12/09/98 02:29 PM

To: David Fallside/Santa Teresa/IBM@ibmus
cc: Ming-Ling Lo/Watson/IBM@IBMUS, ShyhKwei Chen/Watson/IBM@IBMUS, Jen-Yao Chung/Watson/IBM@ibmus, Robert Geiner/Poughkeepsie/IBM@IBMUS, George Zagelow/Santa Teresa/IBM@ibmus
From: Denise Y. Dyko/Poughkeepsie/IBM @ IBMUS
Subject: Re: XML 390 Access Server presentation at the XML summit

Dave, perhaps the word "server" is confusing. Think of the 390 project as an application. I checked with Rob Geiner and he isn't aware that WebSphere is planning on providing XML support for access to legacy data. Are you aware of any WebSphere work in this area? Our application could *work* in a WebSphere environment --we already have a lunchtime meeting scheduled at the XML summit next week to begin discussing implementation environment.

I'm in San Jose this week for the SDC (and to meet with DB2 folks) and have irregular access to the network. Most immediately, my primary interest is getting this topic introduced at the XML summit. Ming and Shyh-Kwei, could you cut down the pitch to 15 minutes? Can you work with Dave and George directly? I suspect I might be a bottleneck this week.

And I'll try to call you, Dave.

Cheers, Denise

Lotus Notes Address: Denise Y. Dyko/Poughkeepsie/IBM@IBMUS

VM Address: KGNVMC.DYKO@VM

Internet Address: DYD@VNET.IBM.COM

Phone: (914)435-6903, t/l 8-295-6903

David Fallside

12/08/98 09:30 PM

To: Denise Y. Dyko/Poughkeepsie/IBM
cc: George Zagelow/Santa Teresa/IBM@ibmus, Jen-Yao Chung/Watson/IBM@ibmus, Ming-Ling Lo/Watson/IBM@IBMUS, ShyhKwei Chen/Watson/IBM@IBMUS
From: David Fallside/Santa Teresa/IBM@IBMUS
Subject: Re: XML 390 Access Server presentation at the XML summit

Denise, after looking over Mingling's foils and reading your reply, I have to say that XAS/390 appears very similar to Websphere and I don't see that XAS/390 is any more or less of an architecture or application than Websphere is already. One option might be

to put a short, say 15 minute, XAS/390 presentation after the Websphere presentation in order to invite questions of comparison.

Another option is to discuss the particular requirements of the 390 platform, perhaps using input from both XAS/390 and the Websphere/390 project.

.....
David C. Fallside, IBM
916.457.2830
fallside@us.ibm.com

Denise Y. Dyko 12/08/98 07:36 AM

To: George Zagelow/Santa Teresa/IBM@ibmus
cc: David Fallside/Santa Teresa/IBM@ibmus, Jen-Yao Chung/Watson/IBM@ibmus, Ming-Ling Lo/Watson/IBM@IBMUS, ShyhKwei Chen/Watson/IBM@IBMUS
From: Denise Y. Dyko/Poughkeepsie/IBM @ IBMUS
Subject: Re: XML 390 Access Server presentation at the XML summit

Hi, George, I don't know the topics of the other proposed presentations but I suspect the 390 application server might be unique in terms of its generalized solution for XML access to legacy data. It's an architecture as much as an application. The first draft of the presentation does not include implementation issues and requirements (should the XML Access Server be implemented as a WebSphere application? under a ComponentBroker environment?). The funding we have does not even cover all of the implementation issues, including currency issues (concurrent access to the data by existing applications and new XML applications). We can add a foil on those issues and requirements.

I expect many of the audience should know about this project and might have requirements for this project, at least in terms of the ability to integrate what they're doing or perhaps in their ability to depend on the existence of this architecture. If necessary, I imagine we can present in half an hour. But I do think it's important that the XML community be aware of this project.

Cheers, Denise
Lotus Notes Address: Denise Y. Dyko/Poughkeepsie/IBM@IBMUS
VM Address: KGNVMC.DYKO@VM
Internet Address: DYD@VNET.IBM.COM
Phone: (914)435-6903, t/ 8-295-6903

 George Zagelow
12/04/98 06:34 PM
.....

To: Denise Y. Dyko/Poughkeepsie/IBM@IBMUS
cc: David Fallside/Santa Teresa/IBM@ibmus
From: George Zagelow/Santa Teresa/IBM@IBMUS
Subject: Re: XML 390 Access Server presentation at the XML summit

Denise, thanks for the proposed talk. We're in evaluation mode as we've got more talks than will fit in the agenda. Is there anything unique in the application? And do you have requirements or issues to bring forward to Robert? I don't see any in the pitch... Thanks much...

Regards, George
Programming Consultant
1-408-463-4041 or 543-4041, FAX 4763
Lotus Notes: George Zagelow/Santa Teresa/IBM@IBMUS

VM: IBMUSM50(ZAGELOW)
Internet: zagelow@us.ibm.com

Denise Y. Dyko 12/04/98 06:48 AM

To: George Zagelow/Santa Teresa/IBM@ibmus
cc: Jen-Yao Chung/Watson/IBM@ibmus, ShyhKwei Chen/Watson/IBM@IBMUS, Ming-Ling Lo/Watson/IBM@IBMUS
From: Denise Y. Dyko/Poughkeepsie/IBM @ IBMUS
Subject: XML 390 Access Server presentation at the XML summit

Hi, George, Ming-Ling Lo or Shyh-Kwei Chen can present at the XML Summit on "XML Access Server for IBM 390." A draft of the presentation is in the attached note (which I also sent you yesterday). I'd suggest 45 minutes for the presentation. The XML Access Server is the XML research project being funded by OS/390.

Cheers, Denise
Lotus Notes Address: Denise Y. Dyko/Poughkeepsie/IBM@IBMUS
VM Address: KGNVMC.DYKO@VM
Internet Address: DYD@VNET.IBM.COM
Phone: (914)435-6903, t/l 8-295-6903

Forwarded by Denise Y. Dyko/Poughkeepsie/IBM on 12/04/98 09:44 AM -----

 Ming-Ling Lo
12/03/98 05:56 PM

To: Denise Y. Dyko/Poughkeepsie/IBM@IBMUS
cc: Jen-Yao Chung/Watson/IBM@IBMUS, ShyhKwei Chen/Watson/IBM@IBMUS
From: Ming-Ling Lo/Watson/IBM @ IBMUS
Subject: XML 390 foils

Hi, Denise:
I am also involved in the XML 390 project.
In the attachment please find the set of foils for tomorrow's phone conference
Please let me know if there is any problem.
Thanks.

-Mingling

Ming-Ling Lo
Research Staff Member, IBM T.J. Watson Research Center
Phone: (914) 784-7734, t/l: 863 7734
Email: mingling@watson.ibm.com, mll@us.ibm.com

Xml390.prz has been deleted (was already in repository My Attachments Repository -> ) from this note on 14 August 2000 by Jen-Yao Chung



Jen-Yao Chung/Watson/IBM

05/27/2004 04:12 PM
This document expires on
05/27/2103

To ShyhKwei Chen/Watson/IBM@IBMUS

cc

bcc

Subject Re: XML Summit Agenda

Jen-Yao Chung
12/10/98 07:06 PM

To: Denise Y. Dyko/Poughkeepsie/IBM@IBMUS
cc: Ming-Ling Lo/Watson/IBM@IBMUS, ShyhKwei Chen/Watson/IBM@IBMUS, Anant
Jhingran/Watson/IBM@IBMUS
From: Jen-Yao Chung/Watson/IBM @ IBMUS
Subject: Re: XML Summit Agenda

Hi Denise, thank to your recommendation. We are glad to see that our xml/390 presentation was accepted.

Ming-Ling, ShyhKwei, please revise the presentation following the instruction for the speaker information. Please run your presentation through Anant first. Thanks. -chung



George Zagelow
12/10/98 02:14 PM

To: Marie Wieck/Somers/IBM@ibmus, Simon Phipps/UK/IBM@IBMGB, Jason Woodard/Armonk/IBM@ibmus, Thomas Rowe/Raleigh/IBM@IBMUS, Ming-Ling Lo/Watson/IBM@IBMUS, Douglas Wright/Somers/IBM@ibmus, Andre Tost/Rochester/IBM@IBMUS, Robert Weida/Thornwood/IBM@IBMUS, Jen-Yao Chung/Watson/IBM@IBMUS, Donald Eastlake/Hawthorne/IBM@IBMUS, Noah Mendelsohn/Lotus, Joe Guthridge/ATL/Lotus@LOTUS, Dan Chang/Santa Teresa/IBM@ibmus, John Ibbotson/UK/IBM@IBMGB, Pat O'Connor/Cupertino/IBM@IBMUS, LOTUS.SBCA8178@VM, Brad Topol/Raleigh/IBM@IBMUS, Rakesh Mohan/Watson/IBM@IBMUS, Andrew Donoho/Austin/IBM@IBMUS, Yih-Shin Tan/Raleigh/IBM@IBMUS, Bob Schloss/Watson/IBM@ibmus, David Fallside/Santa Teresa/IBM@ibmus, David A Epstein/Watson/IBM@IBMUS, sanjiya@watson.ibm.com@IBMUS, Scott Sylvester/Endicott/IBM@IBMUS, David Lection/Raleigh/IBM@IBMUS, Doug Tidwell/Raleigh/IBM@IBMUS, Toby Lehman/Almaden/IBM@IBMUS, Craig Hayman/Raleigh/IBM@IBMUS, Chris Piekny/Toronto/IBM@IBMCA, Dave Pullin/Raleigh/IBM@IBMUS, Tom Glover/Toronto/IBM@IBMCA, Giuseppe Facchetti/Santa Teresa/IBM@ibmus, Stephen Brodsky/Santa Teresa/IBM@IBMUS
cc: Robert LeBlanc/Somers/IBM@IBMUS, Angel Luis Diaz/Watson/IBM@IBMUS, David Sharp/Raleigh/Contr/IBM@IBMUS
From: George Zagelow/Santa Teresa/IBM@IBMUS
Subject: XML Summit Agenda

Greetings, speakers for the XML Summit! Below is the agenda that will be appearing on the web site today. We've tried for some rough groupings of topics, taking into consideration known speaker scheduling conflicts. We'd like to stay as close to this schedule as possible. Please check your names / topics. You'll find the timings to be perhaps shorter than we had discussed. Please try to accomodate, as the schedule is very dense, and we'll have to manage the time closely. If there are issues/changes for speaker names, topic titles, or timing, please let me know asap. In particular, for those talks where multiple presenters are listed, please let me know who the speaker will be. And, finally, the Summit web site has important speaker information, including handling of copies of your presentation. Thanks much, and I look forward to seeing you all next week...

Regards, George
Programming Consultant
1-408-463-4041 or 543-4041, FAX 4763
Lotus Notes: George Zagelow/Santa Teresa/IBM@IBMUS
VM: IBMUSM50(ZAGELOW)
Internet: zagelow@us.ibm.com

Forwarded by George Zagelow/Santa Teresa/IBM on 12/10/98 10:53 AM

Tuesday, December 15th

8:30 Welcome and Introduction - Robert LeBlanc
8:45 Strategy Update - Marie Wieck
9:15 Marketing Update - Simon Phipps
9:45 eBAF - Jason Woodard
10:15 Break
10:45 WebSphere - Tom Rowe
11:30 XML Access Server for the IBM 390 - Ming-Ling Lo
noon Lunch
1:00 XML and Solutions - Doug Wright
1:30 San Francisco - Andre Tost
2:00 RosettaNet, Catalog Architect, and the Information Supply Chain - Robert Weida
2:30 XML/EDI Pilot for IBM Procurement - Jen-Yao Chung
3:00 Break
3:30 Open Trading Protocol (IOTP) - Donald Eastlake
3:50 Lotus - Noah Mendelsohn and Joe Guthridge
4:20 DB2 - Dan Chang
4:50 MQ Series and CICS - John Ibbotson

Wednesday, December 16th

8:30 XML Parser, XPK4J, etc. - Pat O'Connor
9:00 LotusXSL Processor - Scott Boag
9:20 Transcoding and PBC - Brad Topol
9:50 XML/CAF - Rakesh Mohan
10:10 Break
10:40 Trinity - Andrew Donoho
11:00 Host Integration - Yihshin Tan
11:20 IBM Research Topics - Bob Schloss
12:00 Lunch
1:00 Standards Update - Fallside
2:00 BML - David E/ Sanjiva Weerarana (Watson)
2:20 PDML - Scott Sylvester / Doug Boucher (Rochester)
2:40 Lidea - David Lection
3:00 Break
3:30 XML Directory Access and Query - Doug Tidwell
3:50 Modal - Armando Morales (RTP) / Toby Lehman (Almaden)
4:10 Convergence activities - David Epstein or Pat O'Connor
4:20 Visual Age - Craig Hayman
4:50 Component Broker - Chris Piekny

Thursday, December 17th

8:30 XML Java Beans - Dave Pullin
8:50 Net Objects - Tom Glover
9:05 Data Warehousing and XML - Facchetti
9:35 RDF - Bob Schloss for 20 min
9:55 Break
10:35 XMI/RDF/XML Schema positioning - Brodsky
10:55 Summary / Open discussion - Wieck
11:45 Closing - Robert LeBlanc
noon Summit Ends

Exhibit E

XML Access Server for IBM 390

Ming-Ling Lo, Shyh-Kwei Chen, Jen-Yao Chung
Institute of Advanced Commerce
IBM T.J. Watson Research Center

Project Overview

- Enable rapid exploitation of existing enterprise data in XML eCommerce applications
 - Co-existence with existing enterprise data applications
- Store/retrieve XML data in current enterprise data formats
 - Automatic composition of XML docs from and decomposition of XML docs into multiple enterprise data sources
- Partnership with Poukeepsie (Denise Dyko)

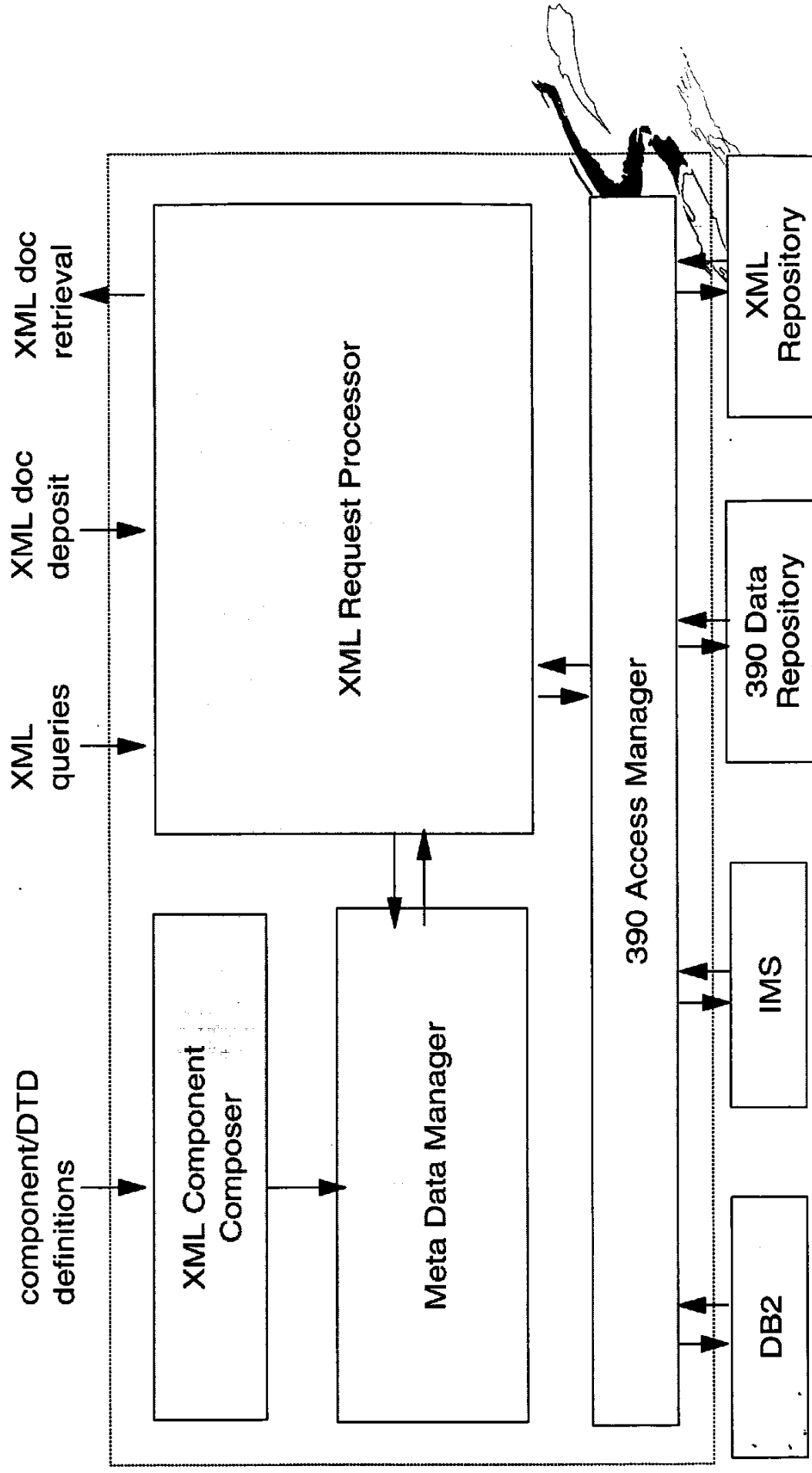
A handwritten signature in black ink, appearing to read 'S. J. Dyko', is located in the bottom right corner of the page.

XML Access Solutions

- A. Store one XML doc in one piece
 - e.g. as one file, as one LOB in a record
- B. Store one XML doc in one or more pieces
 - e.g. linked by xlinks
- C. Store XML docs in XML format
 - tags really appear in physical storage
- D. Store XML docs in more than XML format
 - tags not necessarily appear in physical storage
- XAS Project - An effort in B&D



XAS/390 Overall Architecture

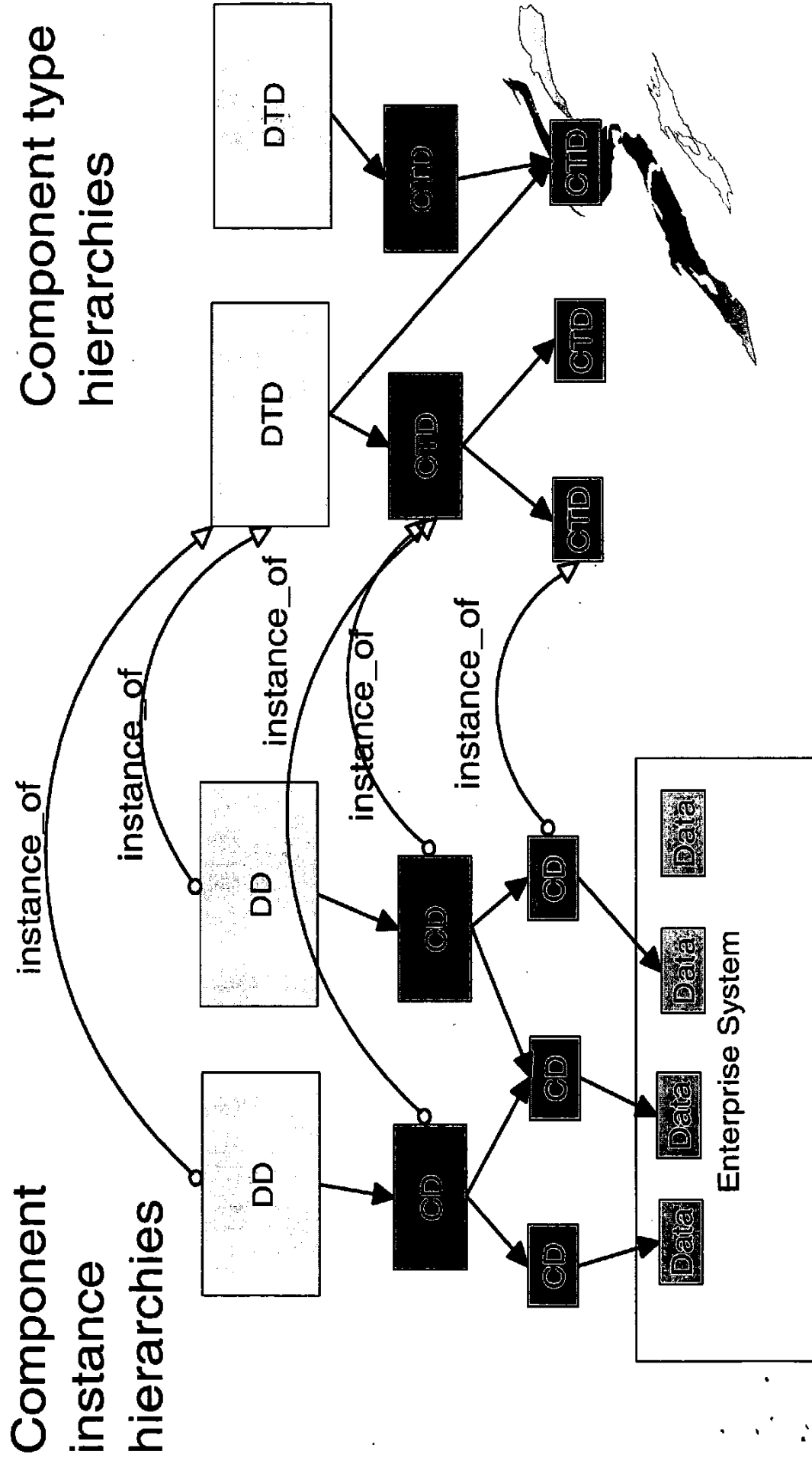


XML Meta Data in XAS/390

- Document type definitions(DTD)
 - Define legal document structures
- Also use the following internal, logical entities ---
- Component type definitions (CTD)
 - Building blocks of DTD
- Document definitions (DD)
 - Instantiation of DTD
- Component definition (CD)
 - Instantiation of CTD
 - Lowest level maps to data sources
- May use XML namespaces, xlink



Meta Data - Component Library

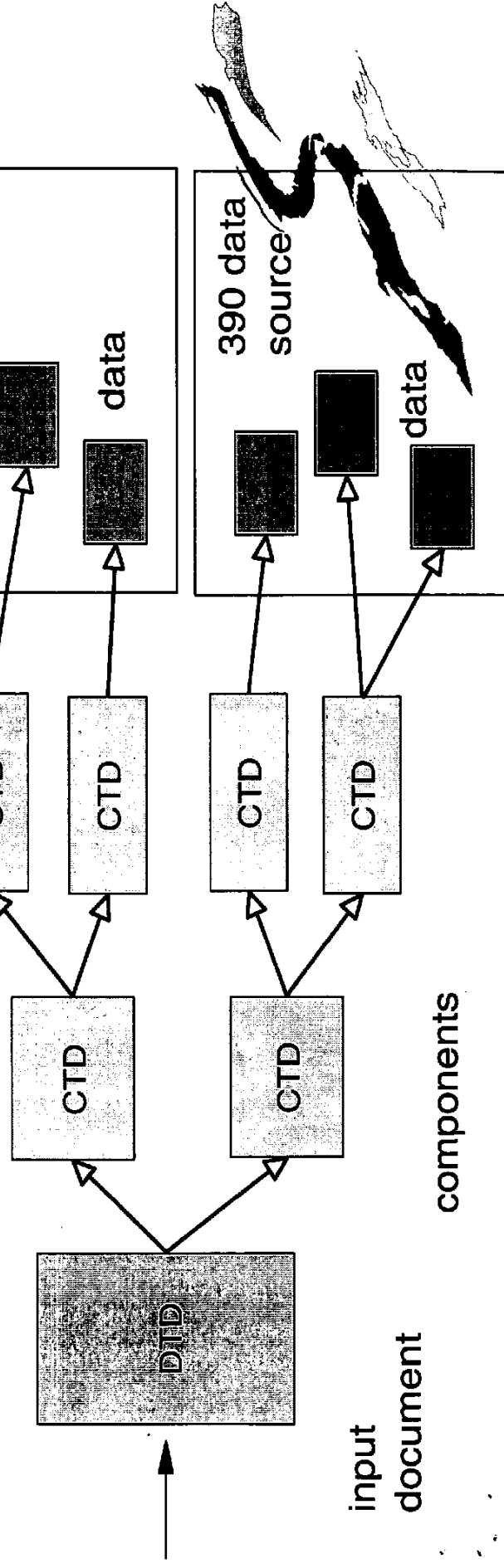


XAS Meta Data Manager

- Consists of
 - XML Meta Data Manager
 - Manage XML component library (DTD, etc..)
 - Manage mapping from CTD/CD to data sources
 - 390 Meta Data Manager
 - Has knowledge about
 - content, locations, formats of enterprise data
 - other information for accessing enterprise data systems (user name, password, etc.)

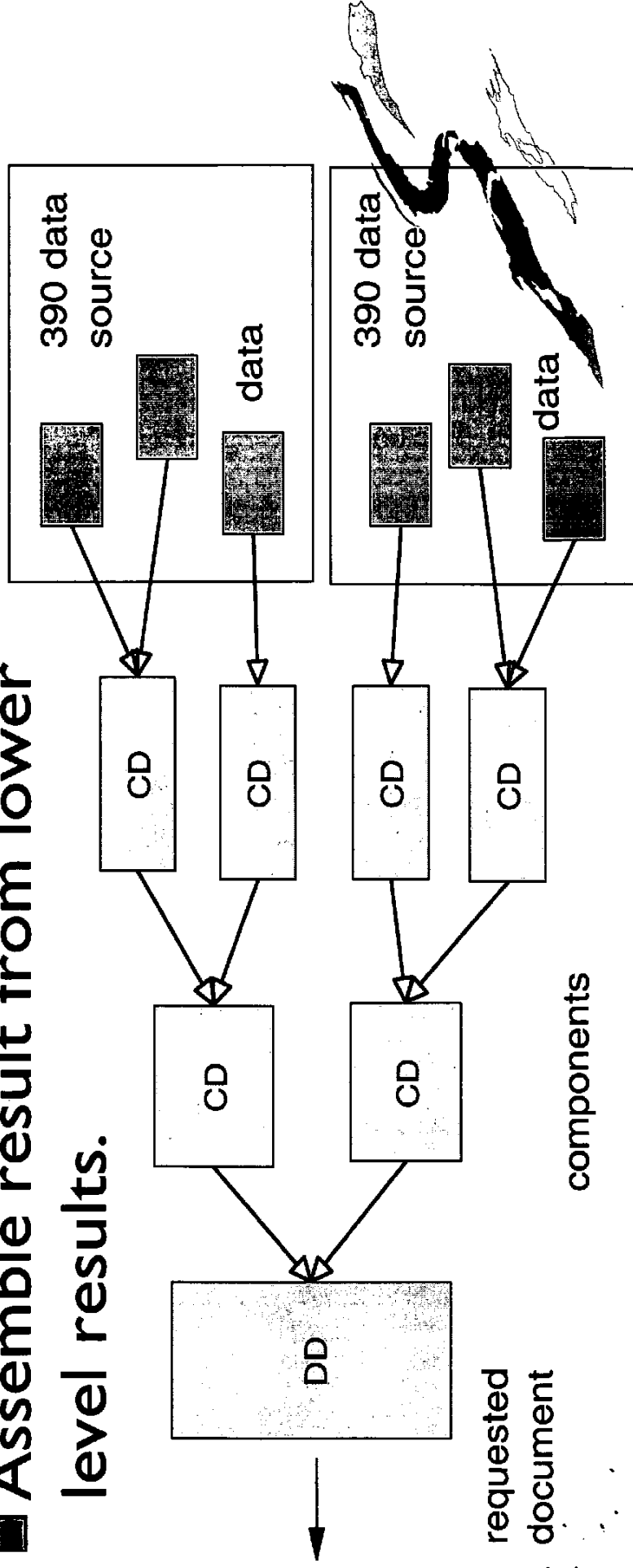
Document Deposit

- Translate into multiple enterprise system source access requests
- Breakup XML doc based on DTD hierarchy



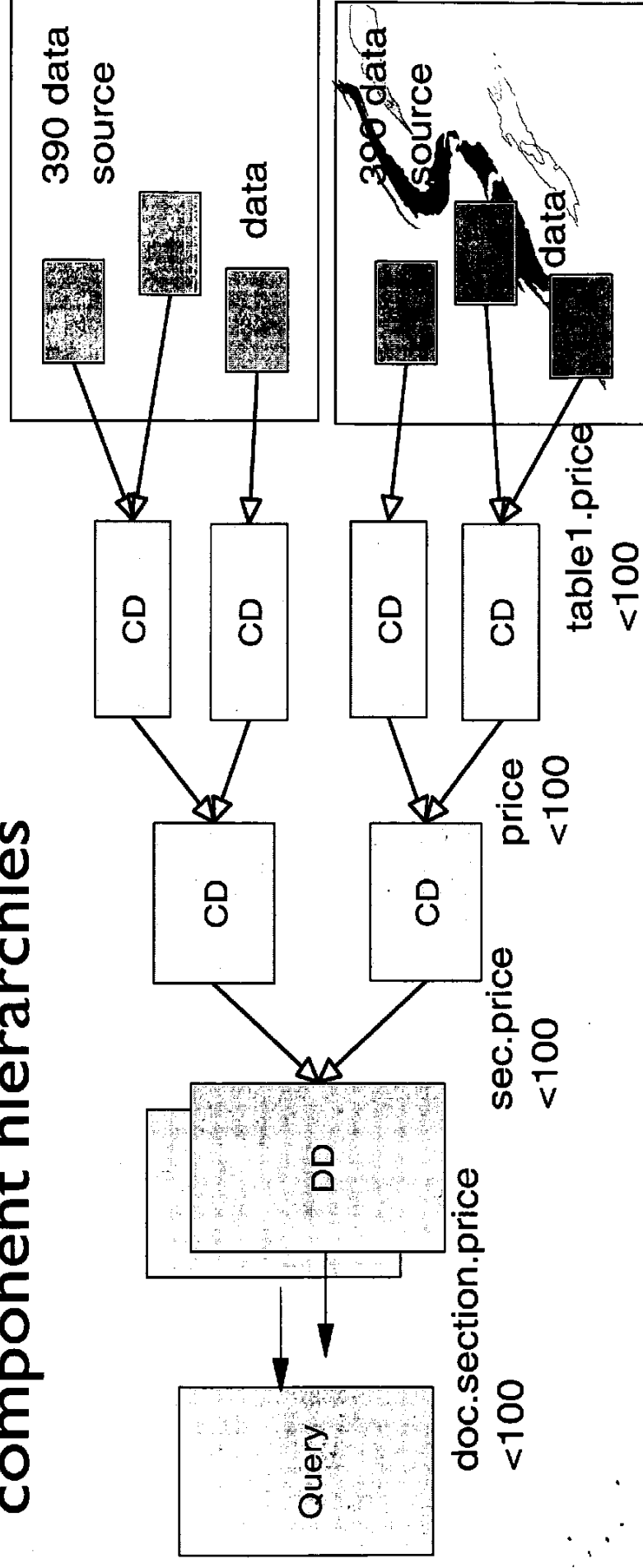
Document Retrieval

- Translated into multiple enterprise system access requests
- Assemble result from lower level results.



XML Query

- Similar to document retrieval
- Push/transform query constraints down component hierarchies



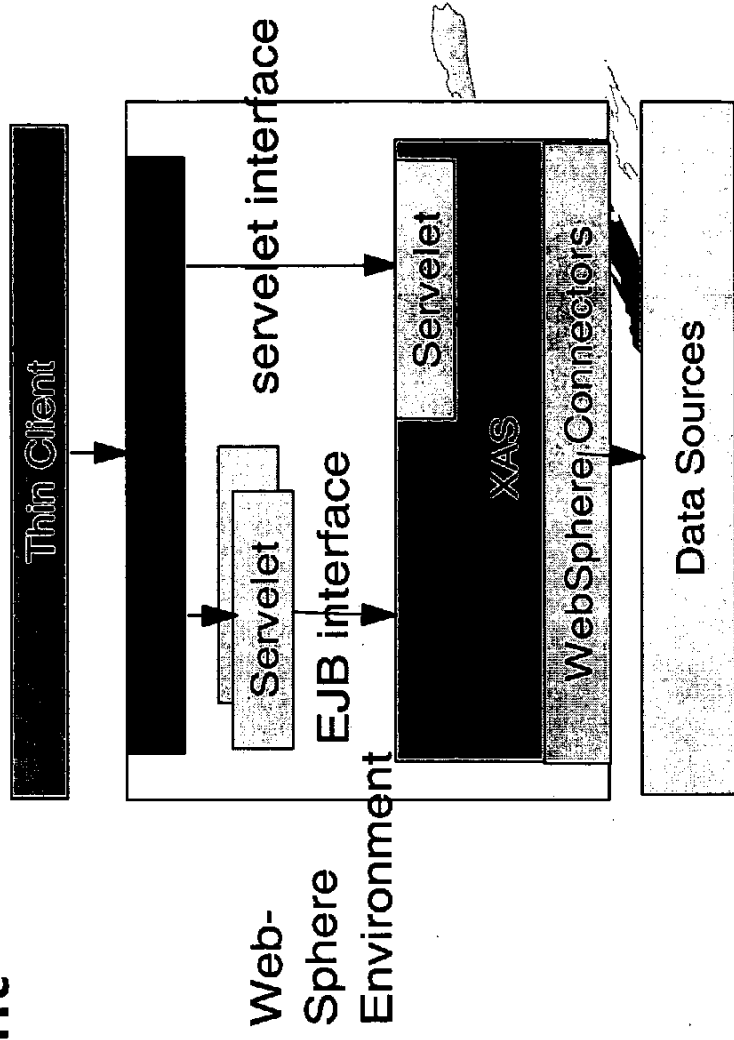
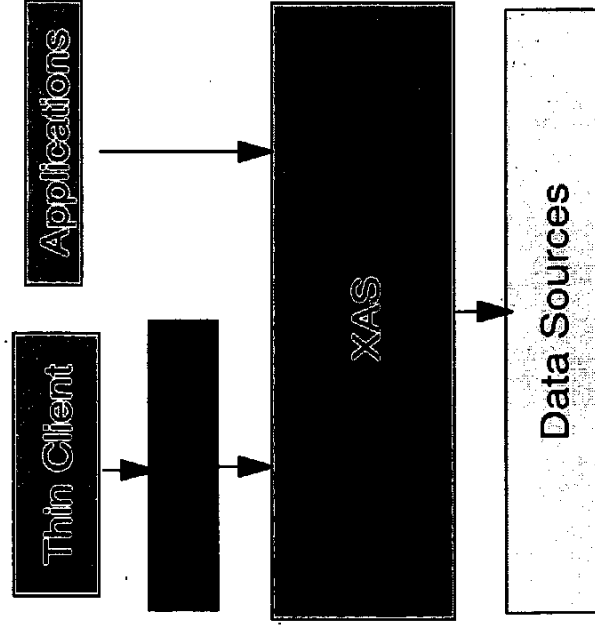
Interesting Research Questions

- Highly dynamic, flexible and cheap doc composition --
 - Create one doc is creating one DD node
- What's important may be doc content but not boundary
 - can we use this to help answer decision support questions?
- Can we dynamically create one big XML doc to answer multiple XML doc query?
- Efficiency Issues
- Data sources not under control



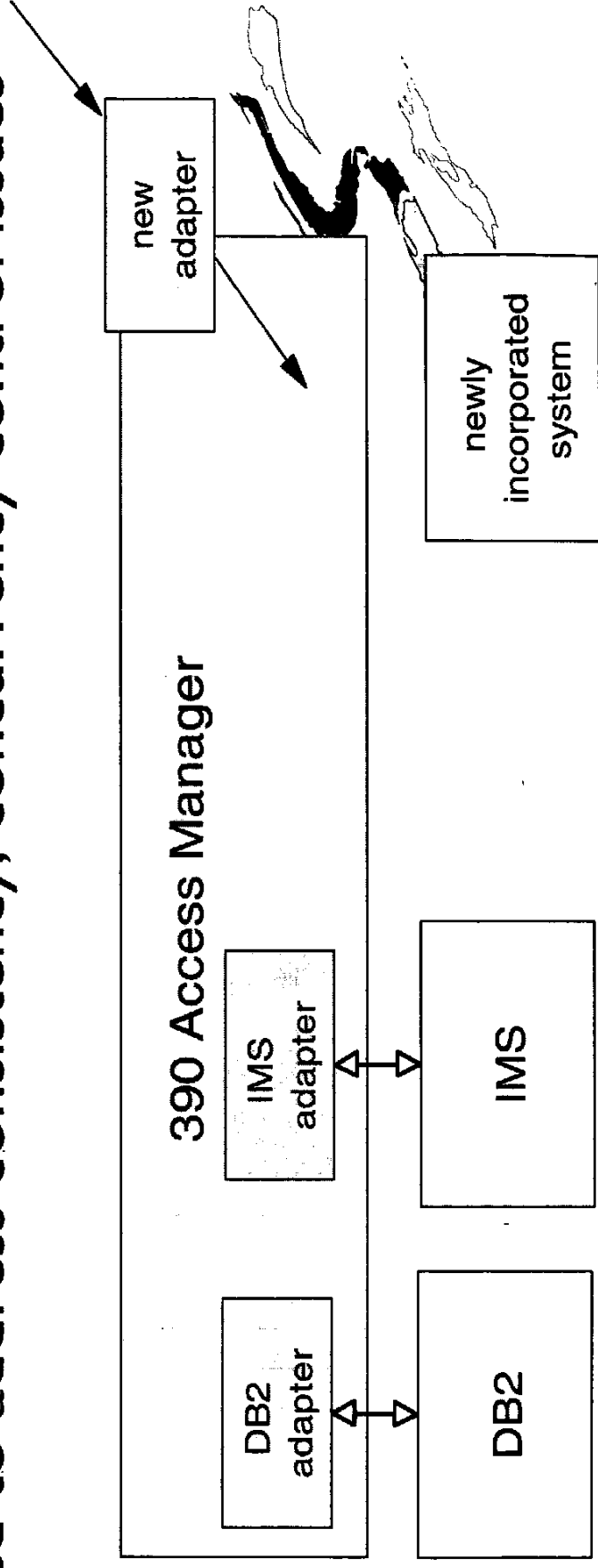
Potential Ways of Application and Deployment

- Standalone
 - Direct access by business applications and browsers.
- In WebSphere environment



390 Access Manager

- ❑ Direct access to 390 enterprise systems
- ❑ Maintain session, connection, etc..
- ❑ Limit data source specific dependency to low level
- ❑ Add new enterprise system by adding adapter
- ❑ Need to address consistency, concurrency control issues



XML Request Processing

- XML document retrievals
 - Translate into enterprise data source retrievals
- XML document deposits
 - Digested and undigested DTD
 - digested DTD: mapping to data sources is defined
 - Initially support only digested DTD
- XML queries
 - Standard not fixed yet
 - Standard may or may not be adequate
 - Closely monitor standard
 - Make implementation flexible, extensible





IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Patent Application Ser. No.: 09/466,627 Group Art Unit: 2176

Filing Date: 12/17/1999

Examiner: M. NGUYEN

Attorney Docket Number YO999-429

Inventor Name(s): LO ET AL.

Title: METHOD AND APPARATUS FOR CONVERTING BETWEEN DATA SETS AND
XML DOCUMENTS

Commissioner for Patents
P.O. Box 1450
Alexandria VA 223131-1450

DECLARATION OF SHYH-KWEI CHEN, PH.D.

Sir:

I, Shyh-kwei Chen, Ph.D., hereby declare as follows:

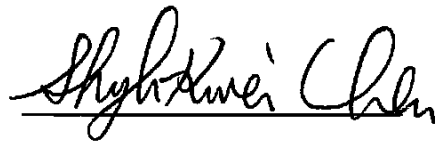
1. I am one of the named inventors in the above-identified application. My co-inventor, Ming-ling Lo, is no longer employed by the assignee of this invention, IBM. Accordingly, he is not readily available and is not joining me on this declaration. Nevertheless, I believe that he would agree with what I declare here.
2. Attached as Exhibit A is a copy of a presentation that I made with my co-inventor Ming-ling Lo and my manager Jen-Yao Chung in December of 1998 discussing the project that Ming-ling Lo and I did relating to XML. This exhibit shows, on page A-3, that we planned to work full time on the project during the entire year of 1999.
3. Attached as Exhibit B is an invention disclosure that I created on April 19, 1999, with my co-inventor Ming-ling Lo, and which matured into the above-identified patent application. I have been informed and believe that this exhibit is a copy of a business

record maintained by the IP Law Department of my employer and assignee, IBM corporation. I therefore trust its accuracy. .

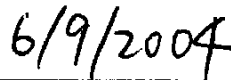
3. Enclosed as Exhibit C is a printout of a computer directory relating to files and/or e-mails relating to this project. These files and/or e-mails have system-generated creation dates that show continuous work on the project that matured into the above-identified patent application from June of 1999 through October of 1999. These documents have titles such as "DTDSA" and "XML," which I recognize as pertaining to this project. These system-generated creation dates are business records that are maintained by the system software of my laptop computer. I do not know how these dates could be altered. I therefore trust their accuracy.
4. Based on looking at Exhibit C, my recollection is refreshed and I also remember that there were earlier files and e-mails dated continuously, throughout the first half of 1999, and relating to this project. These e-mails were destroyed when I got a new laptop on or about June of 1999.
5. Enclosed as exhibit D is a directory printout from my laptop showing the system creation date of exhibit A, namely 12/15/1998.
6. Based on the documents identified above, my recollection is refreshed regarding the events of 1998 and 1999. I therefore remember that Ming-ling Lo and I
 - conceived of the idea of establishing a mapping from lists and scalars corresponding to at least one data source into XML elements and attributes during the summer of 1998;
 - worked at least part time during the fall of 1998 to reduce this invention to practice; and

- worked full time reducing this invention to practice during the entire year of 1999.

7. I further declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issuing thereon.



Shyh-kwei Chen, Ph.D.



Date

Exhibit A

XML Access Server

IBM T.J. Watson Research Center

Ming-Ling Lo

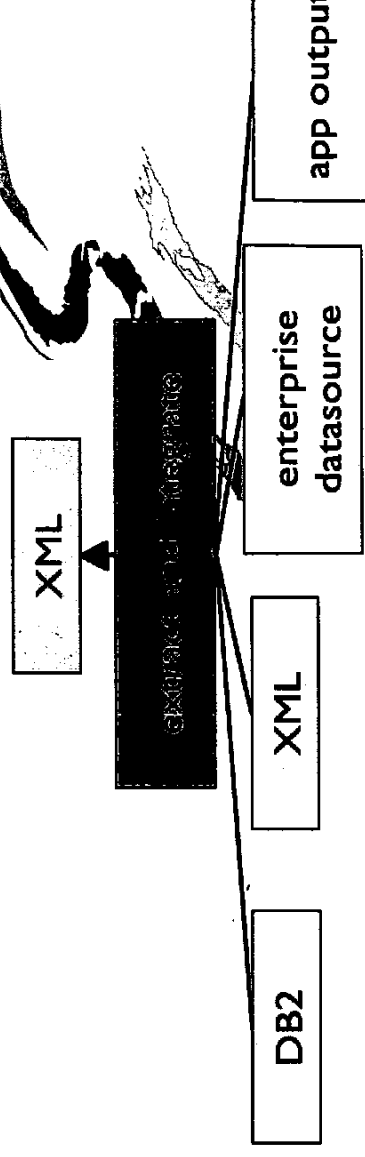
Shyh-Kwei Chen

Jen-Yao Chung



XAS Project Goals

- Machine generated virtual XML documents
- Dynamically extract data from distributed, heterogeneous enterprise data sources, deliver as XML documents
 - Data stored in existing data sources
 - XML as data presentation and transmission format
 - Materialize into XML documents when necessarily
- Motivations: Large amount of data already exists in data systems such as RDB
 - Maximizing investment: re-use data and schema design
 - Avoid re-inventing wheels: business logic already written for RDB, etc; co-exist with existing applications
 - Better data integrity and quality control: one copy of data
 - Connecting to new world: build new eCommerce application on top of XML interface



Project Content

■ Underlying technology:

- Approach: Once per type authoring - one authoring effort per DTD, enable access to hundreds/thousands of XML documents of this DTD
- Rigorous framework for DTD to existing schema mapping
- Algorithms for type-by-type authoring
 - Enterprise data publishing: given existing data, publish as XML
 - Schema layout given DTD: find best table layout for DTD
 - Cross schema matching: Given DTD and RDB schema for similar purpose, establish mapping
 - Componentized XML construction: Use existing data as building blocks of new XML docs

■ Server architecture design and prototyping

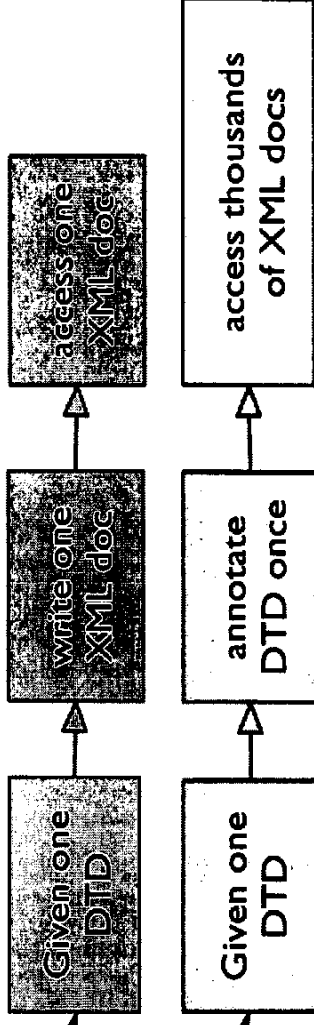
- XML document retrieval, deposit, and query

■ Timeframe: 1-12/1999

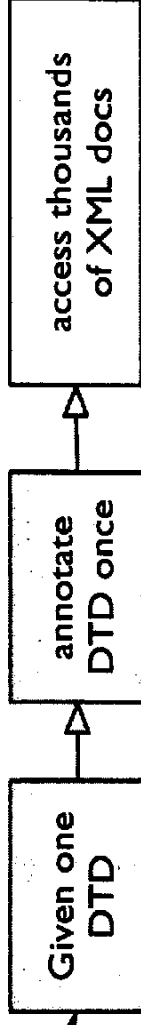


Mapping Technology: DTD-SA

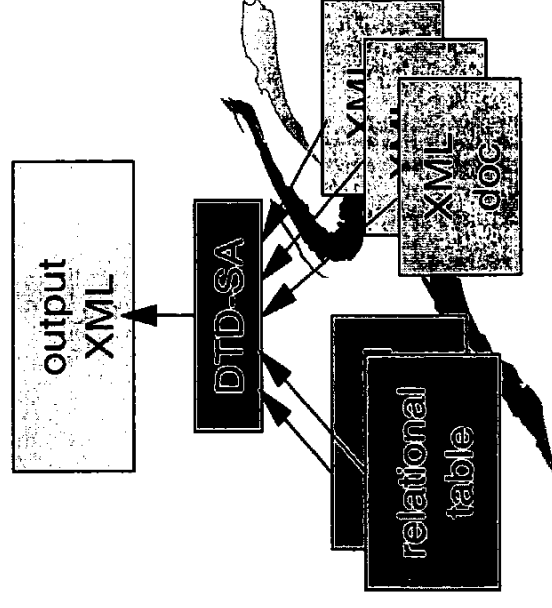
traditional process



DTD-SA process



- DTD-SA = Document Type Definition with Source Annotation
- Framework for specify mapping between DTD and existing schema
 - Existing schema
 - relational-like or DTD-like
- Essentially looks like DTD, but with annotations:
 - value specifications
 - binding specifications
 - Simply DTD when stripped of annotations



Example - Purchase Order

- Purchase order related information described by 4 tables and 1 DTD

PO

POID	BUYER	SELLER
100	20	10

company

COID	NAME	ADDR
10	IBM	NY
20	CITIBANK	NY

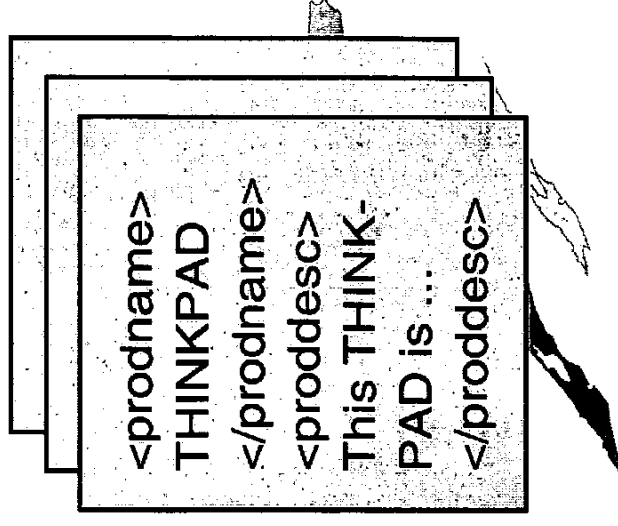
lineitem

POID	PRODID	AMOUNT
100	35678	20k
100	35694	100k

product

PRODID	NAME	DESC.
35678	THINKPAD	
35694	SERVER	

DTD name=pd
(product description)



Example (cont-1)

■ A purchase order DTD

```
<!ELEMENT PO (id, buyer, seller, (lineitem)* )>
<!ELEMENT id (#PCDATA)>
<!ELEMENT buyer (name, address)>
<!ELEMENT seller (name, address)>
<!ELEMENT name (#PCDATA)>
<!ELEMENT address (#PCDATA)>
<!ELEMENT lineitem (prodname, proddesc, amount)>
<!ELEMENT prodname (#PCDATA)>
<!ELEMENT proddesc (#PCDATA)>
<!ELEMENT amount (#PCDATA)>
```


A large, stylized handwritten mark, possibly a signature or a checkmark, is located in the bottom right corner of the page. It is drawn with thick black ink and has a jagged, expressive appearance.

Example (cont-2)

- Add annotation

- blue: value specification
- red: binding specification

```
<!ELEMENT PO (id, buyer, seller, (lineitem)* :: w:= row(lineitem, poid, PO.poid(r)) )>
    :: r:=row(PO, poid, x)
<!ELEMENT id (#PCDATA :PO.<POID>(r) )>
<!ELEMENT buyer (name, address)> :: s:= row(company, id, PO.buyer(r))
<!ELEMENT seller (name, address)> :: s:= row(company, id, PO.seller(r))
<!ELEMENT name (#PCDATA :company.name(s) )>
<!ELEMENT address (#PCDATA :company.addr(s) )>
<!ELEMENT lineitem (
    prodname,
    proddesc :: d:=doc( pd, root.description, product.prodname(v)),
    amount)> :: v:= row( prod, prodid, lineitem.prodid(w))
<!ELEMENT prodname (#PCDATA :product.prodname(v) )>
<!ELEMENT proddesc (#PCDATA :pd.description(d) )>
<!ELEMENT amount (#PCDATA :lineitem.amount(w) )>
```



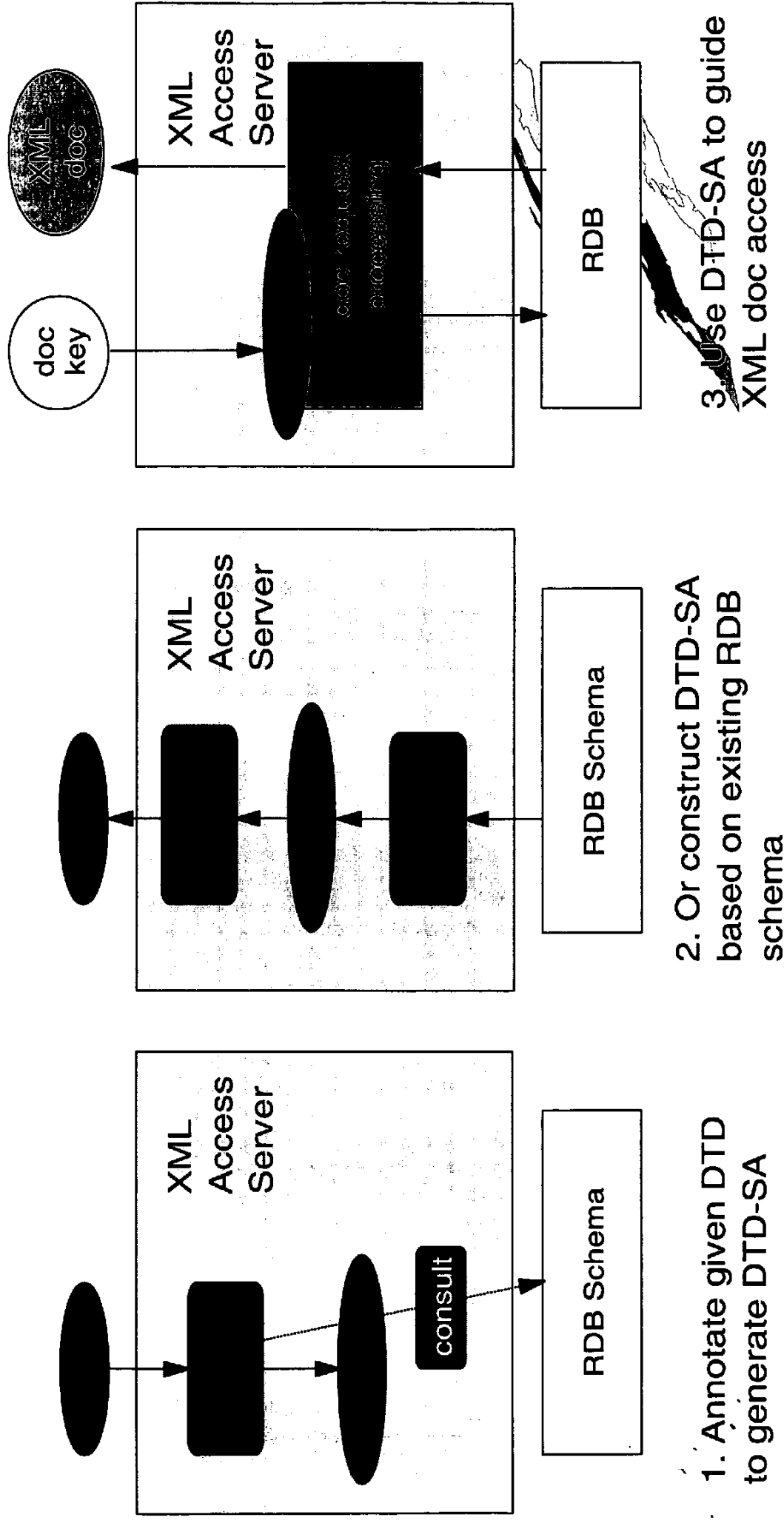
Example (cont-3)

- Retrieve PO document with PO-ID=100 (x=100)

```
<PO>
  <id> 100 </id>
  <buyer>
    <name> CITIBANK </name> <address> NY </address>
  </buyer>
  <seller>
    <name> IBM </name> <address> NY </address>
  </seller>
  <lineitem>
    <prodname> THINKPAD </prodname>
    <proddesc> This THINKPAD is quite good </proddesc>
    <amount> 20K </amount>
  </lineitem>
  <lineitem>
    <prodname> SERVER </prodname>
    <proddesc> This server is the best </proddesc>
    <amount> 100K </amount>
  </lineitem>
</PO>
```



DTD-SA Deployment



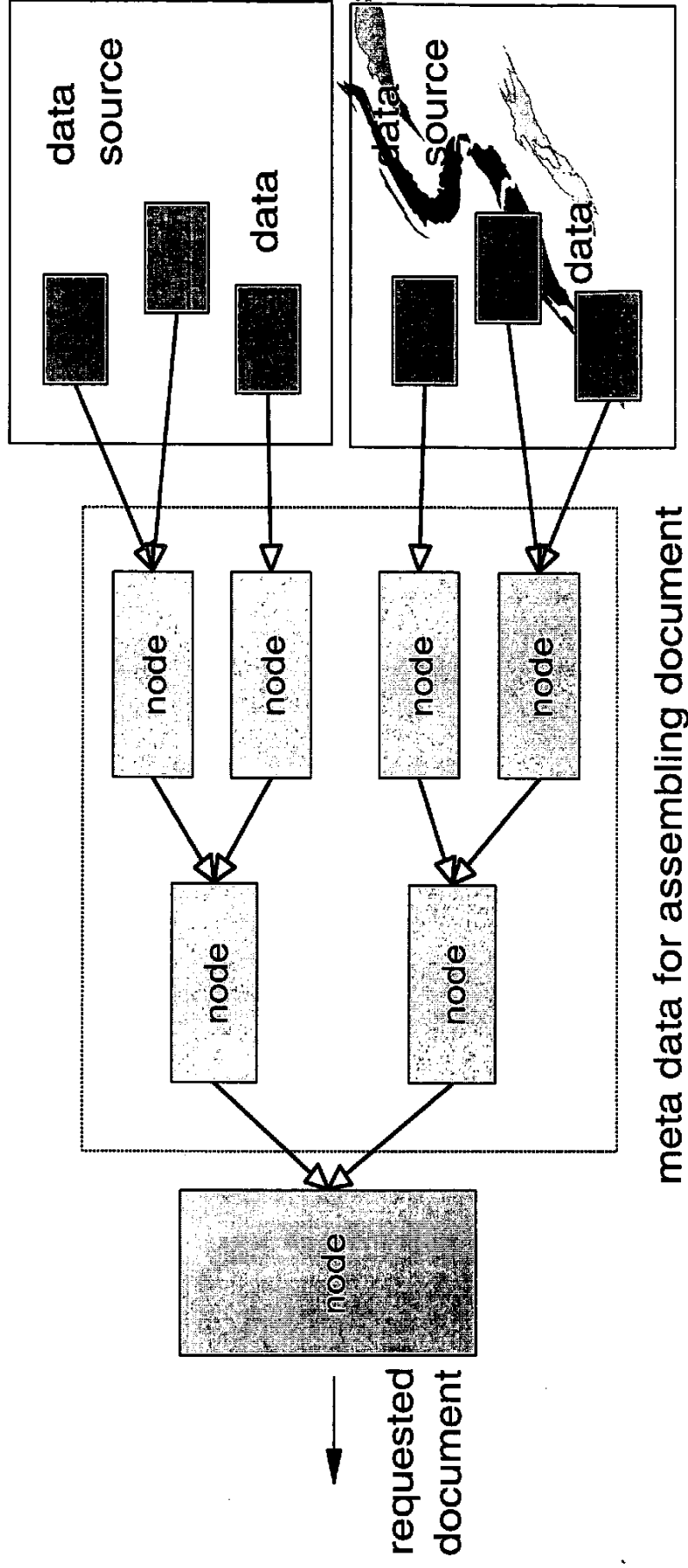
Advantages

- One target DTD can be composed from multiple source tables, schemas, systems (source docs, DTD, repositories)
- Seamless integration of raw data and XML data
 - An output XML document can be woven from mixture of
 - relational columns with raw format data
 - relational columns with XML format data
 - contents of other XML documents
- No changes to source system, schema, and data
- One source schema can map to infinite number of DTDs
- Root element target DTD can start from any source table in a network of the foreign key relationships (any documents in a collection of source XML documents)
- Mathematically rigorous: foundation for automatic algorithms

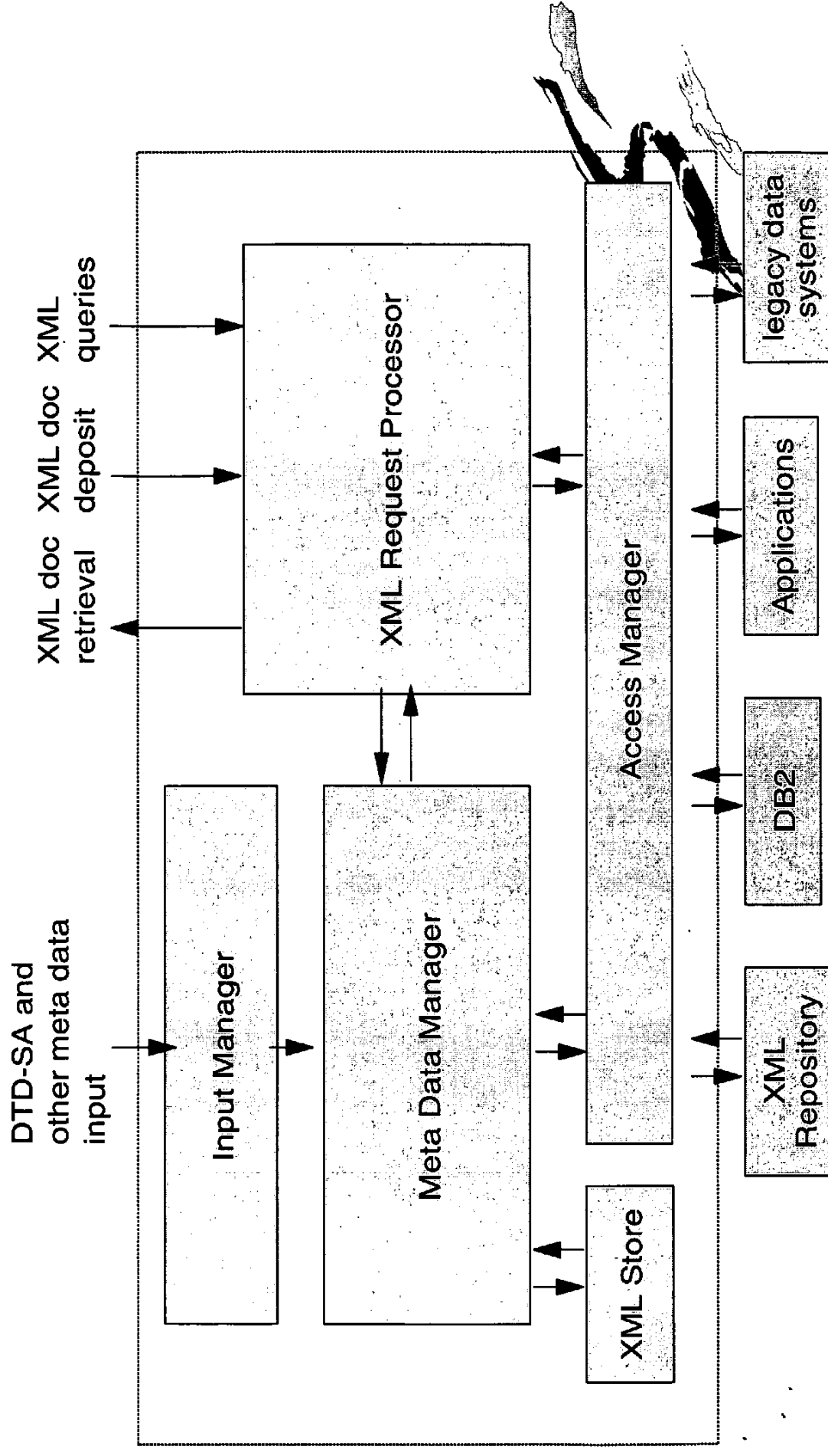


Document Access Implementation

- Doc access translated into multiple enterprise DS accesses
- Assemble result from lower level results.
- Manage meta data for assembling XML doc



XAS Overall Architecture



Project Progress

■ Accomplished

- Developed a new technology for mappings between relational schema and XML DTD, called DTD-SA
- Designed architecture framework
- Ability to extract data from DB2 to XML using DTD-SA

■ Currently under work

- Testing XML extraction using DTD-SA on 390 environment
- Storing XML to single source using DTD-SA
- Investigation 390 specific data source, such as VSAM

■ Next steps

- XML as data sources
- Multiple data sources
- Developing an algorithm for constructing DTD-SA
- Scalable meta data management: DCL Graphs
- Query XML using DTD-SA as underlying mechanism



**Disclosure YOR8-1999-0350**

Created By: Ming-Ling Lo Created On: 04/19/99 05:03:44 PM

Last Modified By: Ming-Ling Lo Last Modified On: 04/21/99 12:01:33 PM

*** IBM Confidential ***

Required fields are marked with the asterisk (*) and must be filled in to complete the form.

Summary

Status	Submitted
Processing Location	YOR
Functional Area	900 Systems & Software-Goyal
Attorney/Patent Professional	Kevin M Jordan/Watson/IBM
Submitted Date	04/21/99 12:00:03 PM
Owning Division	RES
PVT Score	

Inventors with Lotus Notes ID's

Inventors: Ming-Ling Lo/Watson/IBM, ShyhKwei Chen/Watson/IBM

Inventor Name > denotes primary contact	Inventor Serial	Div/Dept	Manager Serial	Manager Name
> Lo, Ming-Ling	013988	22/9AUD	653458	Padmanabhan, Sriram
Chen, Shyh-Kwei	707253	22/9AUF	520894	Chung, Jan-Yao (Chung)

Inventors without Lotus Notes ID's**IDT Selection**

IDT Team:	Attorney/Patent Professional: Kevin M Jordan/Watson/IBM
-----------	------------------------------------------------------------

Main Idea***Title of disclosure (In English)**

A framework for mappings between XML DTD and relational databases

***Idea of disclosure**

1. Describe your invention, stating the problem solved (if appropriate), and indicating the advantages of using the invention.

Problem Description:

XML is emerging as one of the most important format for document and data representation and transmission. Many users and new applications require their input and output to be in XML format. For XML documents, there is the concept of a Document Type Definition (DTD). Each DTD describe the structure of a (potentially infinitely large) set of XML documents. An XML document either has no associated DTD or belongs to exactly one DTD. When an XML document belongs to a DTD, its structure must conform to the specification of the DTD.

There is a large quantity of data already exist in relational databases. These databases are

designed without the expectation that they may one day be accessed for presenting the access result as XML documents. These databases may also have existing applications running on them, still depending on accessing the data in their original format. Furthermore, the data stored in such databases may lack XML specific information such as element tag names or attribute names. Given these facts, it is nonetheless very derivable to access those data in XML format.

High level description of invention:

This invention discloses a framework for creating mappings between XML DTDs and relational database schemas. The relational database can be regarded as having many virtual XML documents of many DTDs stored in them. Each mapping created by our framework specified a subset of these XML documents (which belong to one DTD) explicitly. Given a DTD to the relational database mapping, when a request for a document of this DTD arrives, the request is automatically translated into accesses to various data items in the database. The accessed data items are then assemble to form the requested XML document. The mapping directs which data items to access, and into which element the accessed data item should go.

Advantages:

The advantages of this approach are:

1. Given such a mapping, those who want to access XML document from relational databases no longer need to be concerned with the details of relational databases. All access and query can be expressed in XML terms.
2. Data in relational database need not be explicitly converted into XML format, saving time and space required for such conversion. Also, because only one copy of data is kept, there is no consistency problem between multiple copies of the same data to worry about.
3. It is not necessary to change the relational database system, the relational schema, or the data itself for the need of exporting relational data in XML format. Existing relational database applications can continue to run without modification.
4. The framework supports flexible mappings. In particular, each relational table is not limited to just one "natural" or default mapping. Each table may as many DTDs defined 1. on them as necessary. Also, a DTD is not limited to map to only one relational table. Instead, each DTD can map to multiple relational tables. The foreign key relationship is incorporated into the framework, and expressed seamlessly in the framework.
5. The framework is mathematically rigorous, and can thus be more efficiently and reliably be implemented.
6. The application of this framework is not limited to relational databases. Any data source that can be modeled in a relational manner (such as those expressible by ODBC or JDBC standards) can be used as a data source in our framework.

2. How does the invention solve the problem or achieve an advantage,(a description of "the invention", including figures inline as appropriate)?

Framework for Mappings

This invention defines a set of syntax and binding rules for mappings between XML DTD and relational databases, call DTD-SA (Document Type Definition data Source Annotation)

Notation:

Symbol	Denoting
T	a relational table
C	a column C; assuming each column identifier is unique with the schema
T.C	same as C, but emphasizing the fact that C is a column of table T
Cx, Cy, Cz	Column variables, whose values are column identities
K	a column value
Kx, Ky, Kz	column value variables
<C>	array of columns <C1, C2, ..., Cn>
<K>	an array of column values <K1, K2, ...Kn>. Note: K1, K2, ...Kn may be in different domains.
<<K>>	a sequence of <K>
row(<C>, <K>)	a function defined only when C1, C2, ... Cn belongs to the same table, and Ki is in the domain of Ci, for all i=1,2,...,n. The output of this function is a sequence of rows in table T, with C1=K1, C2=K2, ...Cn=Kn.
T.<C>()	a function which takes a sequence of rows and returns the projection of the sequence in columns <C>
T.C()	shorthand of T.<C>() when there is only one column in <C>

The composite function $T.<C1>(\text{row}(<C2>, <K>))$ can be abbreviated as $T.<C1>(<C2>=<K>)$, or as $T.<C1>(<K>)$ when the identity of <C2> is obvious (e.g. primary key).

Basic Terminology

Specification time: the time when a DTD-SA is specified.

Runtime: the time when a document is either retrieved or queried against.

Basic syntax of DTD-SA

In the following discussion, the function family $F<Cout>(r, n)$, which is a cleaner form for $f(T.<Cout>(r), n)$, where <Cout> is a vector of column identify constants, r is a row variable, and n is an integer, appears many times. The syntax has the following meanings:

1. r is unbounded at specification time and bounded only at runtime.
2. All columns in <Cout> must belong to the same table, T (as is evident from the

syntax $T.<Cout>$).

The sub-construct $r := \text{row}(<C>, <K>)$ also appears many times. The output of $\text{row}(<C>, <K>)$ is a sequence of rows. And the semantics of the construct is:

3. The rows in the sequence are bound to row variable R in turn.

DTD-SA specification rules

The following table lists the original DTD constructs and their annotated counterparts in DTD-SA. For each DTD construct listed a row whose requirement is "must", the construct must be written as the DTD-SA construct as described in the 2nd column. For each DTD construct list in a row whose requirement is "may", it may be replaced with DTD-SA construct listed in the 2nd column, but is not required to. All other DTD construct remains the same in DTD-SA format.

DTD construct	DTD-SA construct	requirement
#PCDATA	#PCDATA :F<Cout>(r, n)	must
#CDATA	#CDATA :F<Cout>(r, n)	must
X, for X=NMToken, NMTOKENS, ID, IDREF, ENTITY, ENTITIES, NOTATION, Enumerated NOTATION.	X:F<Cout>(r, n)	must
...*	...* :: n := F<Cout>(r, m)	must
...*	...* :: r := row(<C>, <K>)	must
...?	...? :: n:=F<Cout>(r, m)	must
(x1x2...xn)	(x1x2...xn) :: n:=F<Cout>(r, m)	must
<!ELEMENT E (...)>	<!ELEMENT E (...)> :: r := row(<C>, <K>)	may

Some terminology can be defined based on the above DTD-SA constructs.

Each DTD-SA construct listed above has either a *content specification* or a *binding specification* associated with it. The content specification is marked by a leading ":", while the binding specification is marked by a leading "::". An content or binding specification is sometimes called a *spec* for convenience. In a DTD-SA, an element definition that contains or is appended with content or binding specifications is called an *annotated element definition (AED)*. Similarly an attribute definition with content or

binding specifications is called an *annotated attribute definition (AAD)*. An AED or AAD is sometime called simply an *annotated definition (AD)* for convenience.

Basic Semantics of DTD-SA Construct

A DTD-SA specification consists of a list of DTD specifications, which when stripped of the content and binding specifications, is simply a DTD.

At the high level, DTD-SA works in the following way: during document retrieval time, a set of row variable bindings is supplied to the DTD-SA, causing all unbounded variables in the DTD-SA to become bounded. In the process, the value, or content, for each element and attribute will be generated. The content of the whole XML document will be generated as a result.

The content specification may have unbounded variables. A content specification denotes that, during document retrieval, the content of its associated element is the output of the function described in the content specification, with all variables, if any, bound to some runtime supplied or derived values.

The binding specification specifies that some unbound row variable in its descendent elements and attributes are bound to the value or expressions listed in the specification. The binding specification generally has the form $r := \text{row}(\langle C \rangle, \langle K \rangle)$, which means the row variable r in the descendent elements will be bound to the output of $\text{row}(\langle C \rangle, \langle K \rangle)$. The output of $\text{row}(\langle C \rangle, \langle K \rangle)$ is a sequence of rows. If the DTD construct being annotated is a regular element definition, the first output row is bound to the element. If the DTD construct being annotated is a "*" construct, r inside the "*" construct and in all its descendent elements will be bound to the output of $\text{row}(\langle C \rangle, \langle K \rangle)$ in turn. The values generated using all these bindings will be the content of the "*" construct.

The specific semantics of the above constructs are further explained below:

a. The integer parameter n :

e.g. $\#PCDATA :f(T.\langle C \rangle(r), n)$

The use of parameter ' n ' is usually for expressing the number of times the annotated $\#PCDATA$ appears in its enclosing construct. The parameter n in $f(T.C(), n)$ is especially useful when $\#PCDATA$ appear inside the * construct.

Note:

i. In most cases $f()$ is actually an identity function.

ii. In the cases $f()$ does not really depend on n , we will shorthand the function as $f(T.\langle C \rangle(r))$. Likewise we can shorthand the function as $f(n)$ or $f()$ when appropriate.

b. $\dots * :f(T.C(r), n):$

in this construct, the output domain of $f()$ is non-negative integers. The output integer m determines the number of repetition for the "*" construct. The numbers 1, 2, ..., m will be bound the unbound variable n insides the "*" construct in turn. The result of these bindings will be the content of the "*" construct.

c. $\dots * ==> \dots * ::r := \text{row}(\langle C \rangle, \langle K \rangle)$

Bind r to the output row($\langle C \rangle$, $\langle K \rangle$) in turn. The number of repetition of the $*$ construct is the number of rows in the output.

d. $\dots? \dots? \dots f(T.C(r), n)$

The output domain of $f()$ is $\{0,1\}$. The output value determine the whether the construct inside the "?" construct occurs once or does not occur.

e. $(x_1x_2\dots x_n) \Rightarrow (x_1x_2\dots x_n) : f(T.C(r), n)$

The output domain of $f()$ is $\{1,2,\dots,n\}$. The output value determines which of the alternatives to take.

f. $\langle !ELEMENT E (\dots) \rangle \Rightarrow \langle !ELEMENT E (\dots) \rangle :: r := \text{row}(\langle C \rangle, \langle K \rangle)$

binding the row variable r to the output of $\text{row}(\langle C \rangle, \langle K \rangle)$. If $\text{row}(\langle C \rangle, \langle K \rangle)$ output more than one rows, r is bound to them in turn. And the content of the construct is a sequence of elements E

Binding rule:

Variables of the same name may appear in various places in a DTD-SA. They may or may not bind to the same value. It is therefore necessary to define the binding rules clearly. There are two types of unbound variables in a DTD-SA, row variables, and integer variables used in determining repetition and alternatives selection.

Distance rules for content and binding specifications:

1. For two specs s_1 and s_2 in the same annotated definition, s_1 is an ancestor spec of s_2 if s_2 is followed, among other symbols, by some number of ")" then by s_1 , in the annotated definition.
2. If s_1 and s_2 are both ancestor specs of s_3 in the same annotated definition, s_1 is closer to s_3 than s_2 , if the number of ")" between s_1 and s_3 is less.
3. If s_1 is defined in annotated definition e_1 , and s_2 defined is annotated definition e_2 . s_2 is an ancestor spec of s_1 if e_2 is an ancestor annotated definition of e_1 .
4. An ancestor spec in the same annotated definition is always closer to an ancestor spec in an ancestor annotated definition.
5. If e_1 and e_2 are ancestor annotated definitions of definition e_3 , and s_1 , s_2 and s_3 are defined inside e_1 , e_2 , and e_3 respectively. s_1 is closer to s_3 than s_2 , if e_1 is closer to e_3 than e_2 .

Binding rules for row variables

6. A row variable r binds to the closest ancestral row binding specification with the same left-hand side (i.e. $:: r := \dots$).

Binding rules for integer variable

7. An integer variable n binds to the closest ancestor integer binding specification with the same left-hand side (i.e. $:: n := \dots$)

How to Access an XML Document using DTD-SA?

To retrieve an XML document using a DTD-SA, one or more row variable bindings must be supplied to the root element, so that all unbound variables in the descent elements and

attributes are bound.

XML retrieval

To access an XML document,

1. (necessary only is DTD-SA system, not needed for single DTD-SA)

Select an element as root element.

2. Supply a "r:= row(<C>, <K>)" to the root element.

Everything else will follow through naturally.

Mixing XML data and raw data:

One great advantage of this method is that the data retrieved from the database can be in either raw format or XML format. The above discussion already explains how data in raw format can be retrieved and put into XML format.

A field in a database table can have either raw (non-XML) data or XML data. Our framework handles both case smoothly. To incorporate data field with XML formatted data, we introduce the two addition DTD-SA specification rules:

DTD construct	DTD-SA construct	requirement
<!ELEMENT E (...)>	<!ELEMENT E (...)> :F<Cout>(r, n)	may
<!ATTLIST E A (...)>	<!ATTLIST E A(...)> :F<Cout>(r, n)	may

The rule for element E means the content of the element E will be replaced by the output of F(), which may contain XML tags. The content of the output of F() is subject to the usual XML parsing and syntax checks. Similar semantics applies to the rule for attribute A.

3. If the same advantage or problem has been identified by others (inside/outside IBM), how have those others solved it and does your solution differ and why is it better?

Comparison with related approaches

There have not been prior inventions related to mappings between XML DTD and relational schema. We therefore compare our invention with the following intuitively conceivable approaches.

Without such a framework for creating mapping between XML DTD and relational schema, there can be the following approaches:

1. For each relational table, assign a default DTD based on the schema of the table. Such an approach is too inflexible. If the DTD is not exactly what the application need, more work is require to convert the DTD to the desired one. Also a DTD is limited to a single table.
2. Issue SQL command, and translate the result to XML format. Again the result format is not flexible enough and has only limited usefulness.
3. Write a potentially complicated SQL program to access the relational database, then fill

the elements of the requested document with the data. The SQL can be complicated, and the logic for filling XML document elements is ad hoc and hardwired in the program.



DTDSA-DATA

Select an item to view its description.

See also:

My Documents

My Network Places

My Computer

Name	Size	Type	Modified
check22	5 KB	File	9/5/1999 12:41 AM
tt22.xml	1 KB	XML Document	9/5/1999 12:41 AM
check2	4 KB	File	9/5/1999 12:40 AM
check5	12 KB	File	9/4/1999 9:46 PM
Test.java	1 KB	JAVA File	9/3/1999 11:44 PM
check	7 KB	File	8/30/1999 3:46 PM
tt4.dtdsa	1 KB	DTDSA File	8/28/1999 11:48 PM
tt5.dtdsa	1 KB	DTDSA File	8/28/1999 11:47 PM
tt3.dtdsa	1 KB	DTDSA File	8/28/1999 1:45 AM
type.txt	5 KB	Text Document	8/28/1999 1:33 AM
tt31.xml	1 KB	XML Document	8/27/1999 1:40 AM
tt21.xml	1 KB	XML Document	8/27/1999 1:40 AM
tt2.dtdsa	1 KB	DTDSA File	8/26/1999 7:34 PM
DtdsaReader.jav...	26 KB	IMP File	8/26/1999 11:32 AM
Depositer.java	6 KB	JAVA File	8/10/1999 5:14 PM
walker.java2	3 KB	JAVA2 File	7/31/1999 12:44 PM
PathToken.java.s...	1 KB	SAVE File	7/31/1999 2:39 AM
Walker.java.save	2 KB	SAVE File	7/30/1999 1:19 PM
pp1.xml	1 KB	XML Document	7/29/1999 12:18 AM
Arbitest.java	2 KB	JAVA File	7/28/1999 5:50 PM
pp.xml	1 KB	XML Document	7/27/1999 6:50 PM
sk-test8.dtdsa	1 KB	DTDSA File	7/26/1999 4:20 PM
sk-test6.dtdsa	2 KB	DTDSA File	7/23/1999 8:39 PM
sk-test4.dtdsa	2 KB	DTDSA File	7/23/1999 8:37 PM
sk-test2.dtdsa	2 KB	DTDSA File	7/23/1999 6:15 PM
sk-test1.dtdsa	2 KB	DTDSA File	7/23/1999 6:11 PM
aa.dtdsa	1 KB	DTDSA File	7/23/1999 2:10 PM
DtdsaReader.jav...	25 KB	SAVE File	7/22/1999 6:11 PM
pp0.dtdsa	1 KB	DTDSA File	7/21/1999 12:52 PM
pc.dtdsa	1 KB	DTDSA File	7/20/1999 5:40 PM
AccessMgr.java.s...	6 KB	SAVE File	7/20/1999 5:06 PM
ProgrammingExce...	1 KB	JAVA File	7/19/1999 1:16 PM
sample01.java	3 KB	JAVA File	7/19/1999 12:42 PM
pp.dtdsa	1 KB	DTDSA File	7/14/1999 5:59 PM

DTDSA-DATA

File Edit View Favorites Tools Help

Back Forward Stop Home Search Folders

Address C:\DTDSA-DATA

Go

DTDSA-DATA

Select an item to view its description.

See also:

My Documents

My Network Places

My Computer

Name	Size	Type	Modified
tt81.xml	1 KB	XML Document	10/10/1999 1:31 AM
tt8.dtdsa	1 KB	DTDSA File	10/10/1999 1:30 AM
tt1.dtdsa	1 KB	DTDSA File	10/9/1999 3:27 PM
tt10.xml	1 KB	XML Document	10/9/1999 3:21 PM
tt20.xml	1 KB	XML Document	10/9/1999 3:21 PM
tt19.xml	1 KB	XML Document	10/9/1999 3:20 PM
tt18.xml	1 KB	XML Document	10/9/1999 3:15 PM
tt17.xml	1 KB	XML Document	10/9/1999 3:15 PM
tt16.xml	1 KB	XML Document	10/9/1999 3:14 PM
tt15.xml	1 KB	XML Document	10/9/1999 3:13 PM
tt14.xml	1 KB	XML Document	10/9/1999 3:13 PM
tt13.xml	1 KB	XML Document	10/9/1999 3:12 PM
tt11.xml	1 KB	XML Document	10/9/1999 3:09 PM
tt12.xml	1 KB	XML Document	10/9/1999 3:09 PM
tt53.check	1 KB	CHECK File	10/8/1999 7:27 PM
tt54.xml	1 KB	XML Document	10/8/1999 7:23 PM
tt53.xml	1 KB	XML Document	10/8/1999 7:21 PM
matchAlgorithm.txt	2 KB	Text Document	10/7/1999 6:49 PM
deposit.txt	5 KB	Text Document	10/1/1999 6:27 PM
deposit.save.txt	5 KB	Text Document	10/1/1999 6:21 PM
tt41.xml	1 KB	XML Document	10/1/1999 4:08 PM
DTDSADP.java	3 KB	JAVA File	9/14/1999 1:42 PM
XMLParser.java	19 KB	JAVA File	9/14/1999 1:42 PM
TraceToken.java	3 KB	JAVA File	9/14/1999 1:42 PM
Node.java	6 KB	JAVA File	9/14/1999 1:42 PM
tt71.xml	1 KB	XML Document	9/13/1999 12:16 PM
tt7.dtdsa	1 KB	DTDSA File	9/13/1999 12:15 PM
tt6.dtdsa	1 KB	DTDSA File	9/13/1999 11:58 AM
tt61.xml	1 KB	XML Document	9/13/1999 11:54 AM
check52	14 KB	File	9/5/1999 1:20 PM
check51	8 KB	File	9/5/1999 2:37 AM
tt52.xml	1 KB	XML Document	9/5/1999 1:00 AM
tt51.xml	1 KB	XML Document	9/5/1999 12:49 AM
check22	5 KB	File	9/5/1999 12:41 AM

349 KB My Computer

**DTDSA-DATA**

Select an item to view its description.

See also:

[My Documents](#)[My Network Places](#)[My Computer](#)

Name	Size	Type	Modified
XMLParser.class	10 KB	CLASS File	10/14/1999 2:02 PM
TraceToken.class	3 KB	CLASS File	10/14/1999 2:02 PM
TokenReader.class	2 KB	CLASS File	10/14/1999 2:02 PM
Test.class	1 KB	CLASS File	10/14/1999 2:02 PM
sample01.class	2 KB	CLASS File	10/14/1999 2:02 PM
Retriever.class	17 KB	CLASS File	10/14/1999 2:02 PM
ProgrammingExce...	1 KB	CLASS File	10/14/1999 2:02 PM
Node.class	5 KB	CLASS File	10/14/1999 2:02 PM
DtdsaReader.class	12 KB	CLASS File	10/14/1999 2:02 PM
DTDSADP.class	2 KB	CLASS File	10/14/1999 2:02 PM
DTDSA.class	2 KB	CLASS File	10/14/1999 2:02 PM
Depositer.class	4 KB	CLASS File	10/14/1999 2:02 PM
Arbitest.class	2 KB	CLASS File	10/14/1999 2:02 PM
AccessMgr.class	5 KB	CLASS File	10/14/1999 2:02 PM
DTDSA.java	3 KB	JAVA File	10/14/1999 2:01 PM
Retriever.java	40 KB	JAVA File	10/14/1999 11:46 AM
AccessMgr.java	8 KB	JAVA File	10/14/1999 12:42 AM
test9.dtdsa	2 KB	DTDSA File	10/13/1999 4:33 PM
test8.dtdsa	1 KB	DTDSA File	10/13/1999 4:33 PM
test7.dtdsa	1 KB	DTDSA File	10/13/1999 4:33 PM
test6.dtdsa	2 KB	DTDSA File	10/13/1999 4:33 PM
test5.dtdsa	2 KB	DTDSA File	10/13/1999 4:33 PM
test4.dtdsa	2 KB	DTDSA File	10/13/1999 4:33 PM
test2.dtdsa	2 KB	DTDSA File	10/13/1999 4:33 PM
test11.dtdsa	1 KB	DTDSA File	10/13/1999 4:33 PM
test10.dtdsa	2 KB	DTDSA File	10/13/1999 4:33 PM
test1.dtdsa	2 KB	DTDSA File	10/13/1999 4:33 PM
DtdsaReader.java	26 KB	JAVA File	10/13/1999 4:21 PM
exp.txt	1 KB	Text Document	10/11/1999 10:51 AM
check.txt	7 KB	Text Document	10/11/1999 2:47 AM
check2.txt	7 KB	Text Document	10/11/1999 2:46 AM
ttt81.xml	1 KB	XML Document	10/10/1999 1:31 AM
ttt8.dtdsa	1 KB	DTDSA File	10/10/1999 1:30 AM
ttt1.dtdsa	1 KB	DTDSA File	10/9/1999 3:27 PM

	Filename	Date	Time	Orig Size	Comp Size	Method	Attr	CRC32	Ratio
1	DCL creation.lwp	4/19/1999	7:38:26 pm	83,836	34,751	DeflatedN	w	4479f60d	58.6%
2	DTDSA patent.lwp	8/24/1999	6:36:02 pm	107,864	42,809	DeflatedN	w	d654bbe3	60.4%
3	Mapping patent.PRZ	5/13/1999	7:42:34 pm	112,601	23,078	DeflatedN	w	d0c8403a	79.6%
4	XAS Joint Program Review 7-	7/12/1999	8:03:12 pm	96,463	20,957	DeflatedN	w	78ae2304	78.3%
5	Xml390-talk.prz	12/15/1998	3:27:20 am	90,470	19,226	DeflatedN	w	f28d2894	78.8%
6	390xmlwkbk.lwp	9/30/1998	2:14:54 pm	50,692	23,766	DeflatedN	w	29d90d0d	53.2%

6 files, 541,926 bytes

RELATED PROCEEDINGS APPENDIX

None¹

¹ Please note request in brief to suspend rules so decision on petition formerly included in this appendix can be reviewed.